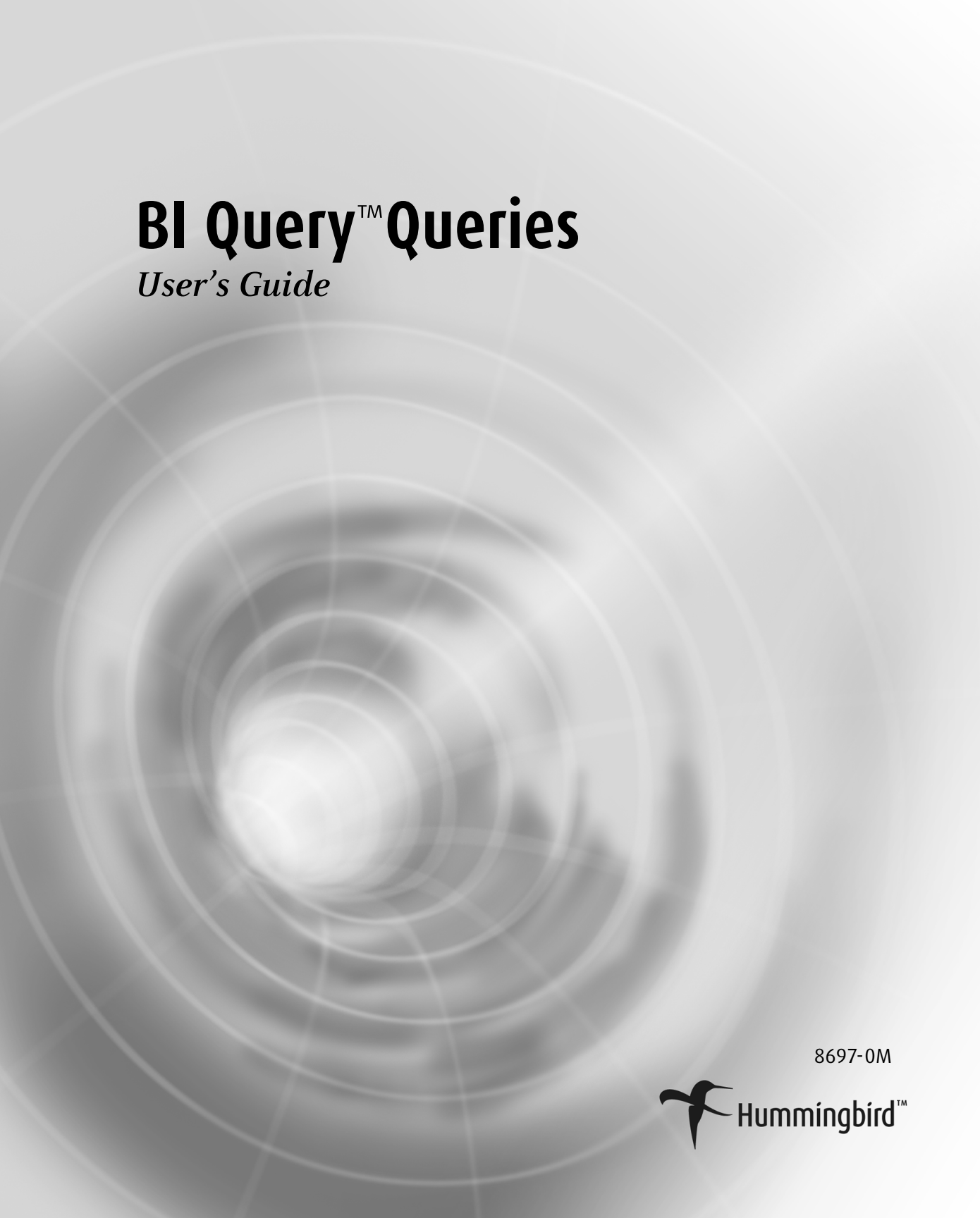
BI Query™ Queries

User's Guide



8697-0M

Hummingbird™

BI Query Queries User's Guide
0800 8697-0M
5/31/02

Hummingbird Ltd.
1 Sparks Avenue, Toronto, Ontario, Canada M2H 2W1
Tel: +1-416-496-2200 Toll Free Canada/USA: 1-877-FLY-HUMM (1-877-359-4866)
Fax: +1-416-496-2207
E-mail: support@hummingbird.com or getinfo@hummingbird.com
FTP: ftp.hummingbird.com
For more information, visit www.hummingbird.com

RESTRICTED RIGHTS LEGEND. Unpublished rights reserved under the copyright laws of the United States. The SOFTWARE is provided with restricted rights. Use, duplications, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (c) (1) and (2) (a) (15) of the Commercial Computer Software-Restricted Rights clause at 48 CFR 52.227-19, as applicable, similar clauses in the FAR and NASA FAR Supplement, any successor or similar regulation.

Information in this document is subject to change without notice and does not represent a commitment on the part of Hummingbird Ltd. Not all copyrights pertain to all products.

© 1988–2002 Hummingbird Ltd. All rights reserved.

Genio, Genio Suite, Genio Miner, Genio MetaLink, Genio MetaLink for SAP R/3, Genio MetaLink for ERwin, Genio MetaLink for Power Designer, Genio MetaLink for IDOC, Genio MetaLink for BW, Genio Designer, Genio Scheduler, Genio Administration Manager, Genio Engine, Genio Scheduling Service, Genio Polling Service, Genio BW Service, Genio Met@Data, Hummingbird BI, BI Analyze, BI Cube Creator, BI Server, BI Query, BI Query Admin, BI Query Reports, BI Query Chart Editor, BI Query Update, BI Query User, BI Web Personal Portfolio, BI Web, Hummingbird Portal, and Hummingbird Core Services are trademarks of Hummingbird Ltd. and/or its subsidiaries.

ACKNOWLEDGEMENTS WinHelp extensions copyright © Blue Sky Software Corporation. This product makes use of the Blowfish library, an SSL implementation written by Eric Young, © 1995–1997, Eric Young. Photo clipart copyright © 1996 PhotoDisc Inc. All rights reserved. All other copyrights, trademarks, and tradenames are the property of their respective owners.

DISCLAIMER Hummingbird Ltd. software and documentation has been tested and reviewed. Nevertheless, Hummingbird Ltd. makes no warranty or representation, either express or implied, with respect to the software and documentation included. In no event will Hummingbird Ltd. be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect in the software or documentation included with these products. In particular, Hummingbird Ltd. shall have no liability for any programs or data used with these products, including the cost of recovering such programs or data.

Related Documentation and Services

Manuals

All manuals are available in print and online. The online versions require Adobe Acrobat Reader 5.0 and are installed only if you do a Complete installation. Your Hummingbird product comes with the following manuals:

BI Query Installation Guide	Determine system requirements and install BI Query and BI Query Reports.
BI Query Queries User's Guide	Query corporate databases and export data to other applications.
BI Query Data Models User's Guide	Create and manage data models and update records in the database.
BI Query Reports User's Guide	Produce reports using BI Query Reports from data obtained using BI Query.
BI Query Chart Editor User's Guide (PDF Only)	Use advanced features to edit charts created in BI Query Reports.

Help

The online Help is a comprehensive, context-sensitive collection of information regarding your Hummingbird product. It contains conceptual and reference information, and detailed, step-by-step procedures to assist you in completing your tasks.

Release Notes

The release notes for each product contain descriptions of the new features and details on release-time issues. They are available in both print and HTML. The HTML version can be installed when you install the software. Read the release notes before installing your product.

Professional Services

Hummingbird offers consulting and training services worldwide. Working alongside your technical and non-technical staff, Professional Services can help you identify areas where improved information management can enhance your business performance. As well, we can provide training on how to use your Hummingbird products. If requested, we can design courses that are tailored to meet your organization's specific needs. These courses can take place at your workplace or at our own training centers. To register, or for more information, pricing, and detailed course outlines, contact Hummingbird Professional Services.

Telephone	+1-613-548-4355 ext. 1700
Fax	+1-613-548-7801
Email	proservices@hummingbird.com
Website	www.hummingbird.com

Hummingbird Exposé Online

Hummingbird Exposé Online is an electronic mailing list and online newsletter. It was created to facilitate the delivery of Hummingbird product-related information. It also provides tips, help, and interaction with Hummingbird users. To subscribe/unsubscribe, browse to the following web address:

```
http://www.hummingbird.com/expose/about.html
```

User Groups and Mailing Lists

The user group is an unmoderated, electronic mailing list that facilitates discussion of product-related issues to help users resolve common problems and to provide tips, help, and contact with other users.

To join a user group:

Send an e-mail to `listserv@hummingbird.com`. Leave the Subject line blank. In the body of the e-mail message, type the following:

```
subscribe hbi-users Your Name
```

To unsubscribe:

Send and e-mail to the listserv address. Leave the Subject line blank, and type `unsubscribe hbi-users` in the body of the e-mail message.

To post messages to the user group:

Send your e-mail to:

```
hbi-users@hummingbird.com
```

To search the mailing list archives:

Go to the following web site:

```
http://www.hummingbird.com/support/usergroups.html
```

Contents

Chapter 1: Introducing BI Query 1

About BI Query 3
- BI Query Admin, User, and Update 3

Basic Concepts 4
- Database Components 4
- Data Model Components 4
- Queries 7
- Reports 8
- Types of Data Models 8
- BI Query Operation Modes 11
- Using BI Query with BI Server 12

Chapter 2: Getting Started 15

Starting BI Query 17
- Opening Data Models 18
- Retrieving Data Models 19
- Finding the Administrator Layer in a Split Data Model 20
- Working Online or Offline 21

Connecting to a DBMS 21
- Disconnecting from the DBMS 23

Chapter 3: Building Queries 25

Query Basics 27
- Building Queries 28
- Submitting and Clearing Queries 29
- Cancelling Queries 30
- Reordering Columns Before Submitting a Query 31
- Sorting Rows 31

Viewing Query Specifications 33
Saving Queries 33
Opening Saved Queries 35
Qualifying Queries 36
Using the Qualification Tree 36
Adding, Modifying, or Deleting Qualifications 38
Qualifying with Available Data Values 39
Applying Multiple Qualifications with And/Or 41
Combining Qualifications 42
Using Negative Qualifications 42
Using Query Modifiers 43
Qualifying One Attribute with Another 44
Using Qualification Operators 45
Matching Character Patterns 46
Using Wildcard Characters 46
Making Comparisons 47
Matching Items in a List 47
Finding a Range of Values 48
Matching Any/Matching All 48
Determining if Data is Missing 49
Applying Your Own Operators 50
Querying Multiple Data Objects 50
Removing Unnecessary Joins from Queries 51
Updating Queries 52
Refreshing a Query 52
Using Legacy Joins As Is 53

Chapter 4: Formatting Data 55

About Data Formats 57
Data Type Categories 57
Data Format Options 58
Format Order 59
Formatting Data: The Basics 60
Opening the Format Dialog Box 61
Specifying Data Model Default Formats 63
Setting the Null Values Option 64
Setting the String Option 65
Setting the Computed String Option 66

Formatting Specific Data Types 68
Formatting Boolean Data 68
Formatting Numeric Data 69
Formatting Date and Time Data 71
Formatting Data for Data Entry 72
Setting the Data Entry Format 73
Entering Date and Numeric Data 73
Converting Year Formats 74
Correcting Invalid Entries 75

Chapter 5: Using Data Values Files 77

About Data Values Files 79
Types of Data Values Files 80
Naming and Storing Data Values Files 80
Order of Data Values File Use 81
Creating Data Values Results Files 82
Creating Data Values Results Files Using a Text Editor 83
Creating Data Values Query Files 85
Using a Data Values File in a Query 86
Data Aliasing 87
Creating a Data Values File to Alias Data 88

Chapter 6: Using Prompts 89

About Prompts 91
Types of Prompts 92
Naming Conventions for Prompts 94
Data Types for Prompts 94
Required Versus Optional Prompts 96
Prompts for BI Web Reports 97
Common Tasks Using Prompts 99
Creating a Single Prompt 100
Inserting a Prompt in a Query 102
Creating a Group Prompt 103
Testing a Group Prompt 103
Creating a Conditional Prompt 105
Testing a Conditional Prompt 105
Importing Prompts 106
Showing Prompt Values in Queries 107
Specifying Web-Reporting Values 107

Chapter 7: Using Variables **111**

About Variables 113
Typical Uses for Variables 114
Expressions for Variables 114
Using Literals in Variables 114
Using Placeholders in Variables 115
Using Functions, Prompts, and Other Values in Variables 116
Creating and Using Variables 117
Using Variables to Qualify an Attribute 119
Importing Variables from a File 120
Using Variables to Supply User or Group Names 121

Chapter 8: Working with Results **123**

About Results 125
Opening Results 126
Modifying a Query for Selected Results 127
Using Super Queries 129
About the Super Query Window 129
Opening the Super Query Window 132
Guidelines for Editing Super Queries 132
Editing Component Queries 133
Editing an Operation in the Super Query Window 134
Correcting Errors 135
Filtering Results 136
Getting Results in Sorted Order 136
Reordering Columns 137
Narrowing the Range of Results 139
Combining Results Sets 140
About Appended Columns 141
Combining Results by Appending Rows 142
About Column Joins 143
Types of Column Joins 144
Combining Results by Joining Columns 146

Chapter 9: Creating Standard Reports 149

About Standard Reports 151
- Standard Report Structure 152
- Data Formats in Standard Reports 153

Basic Report Tasks 153
- Creating Standard Reports 153
- Saving Report Specifications 154
- Applying Report Specifications 156
- Printing and Exporting Standard Reports 157

Adding Calculations to Standard Reports 157
- Adding Subtotals 158
- Adding Grand Totals 159
- Adding Calculated Columns 160

Manipulating Columns in Standard Reports 162
- Resizing Report Columns 162
- Combining Columns 163
- Banding Columns 164
- Stacking Columns 165
- Formatting Column Text and Headings 166
- Customizing Column Settings 167

Adding Borders to Standard Reports 168

Using Ornaments in Standard Reports 169
- Special Text Strings in Standard Reports 170
- Adding Text Ornaments to Standard Reports 171
- Editing Default Ornaments in Standard Reports 172

Converting Standard Reports 173
- Converting Single Standard Reports 174
- Converting Multiple Standard Reports 174
- Setting Batch Converter Options 175

Chapter 10: Building Advanced Queries 177

Applying Functions to Attributes 179
- Built-In Aggregate Functions 180
- Types of Grouping 181
- Applying a Function After Grouping 182
- Applying a Function Before Grouping 184
- Grouping without Functions 185

Creating Calculated Attributes 186
Using Functions in Query Expressions 189
Using Subqueries 190
Creating and Inserting Subqueries 191
Using Data Sets 192
Data Set File Syntax 193
Using a Data Set to Qualify an Attribute 194
Viewing Data Sets 194
Using a Data Values Results File as a Data Set 195
Creating Tables in the Database from Query Results 196
Creating Dynamic Relationships 197
Deleting a User-Created Relationship 201
Using Outer Joins in a Query 201
Specifying Join Order 203
About Join Conflicts 204
About Outer-Join Syntax in Queries 206
Changing Outer-Join Syntax for a Connection 206
Modifying the SQL for a Query 207
Using SQL Prefixes and Suffixes 209
Building Circular Queries 210
Circular Queries with a Correlation 210
Building a Circular Query with a Correlation 213
Circular Queries without a Correlation 214
Building a Circular Query without a Correlation 215

Chapter 11: Exporting Information 217

Exporting Information 219
Preparing Results for Export 219
Results Options and Separators 220
Specifying Results Options and Separators 222
Export Options 223
Setting Export Options 224
Export Methods 226
Exporting Results to a Text File 226
Exporting Results to an Application Using a Button 227
Exporting Results Using E-Mail 229
Exporting the SQL for a Query 230

Exporting Results to Microsoft Excel 230
Using Two DDE Commands 231
Opening Macro Sheets Automatically 231
DDE Command Examples for Excel 232
Combining Multiple Results Sets 233

Chapter 12: Working with BI Server 237

Working with BI Server 239
Requirements for Publishing Queries 239
Publishing Queries 239
Retrieving Queries 241
Deleting Published Queries 241
Publishing Standard Report Specifications 241
Publishing Data Values Results Files 243
Controlling Access Using BI Server 243
Users and Groups in BI Server 244
Access Inheritance Rules 244
About the Set Security Dialog Box 245
Setting Security 246
Setting Security for Queries 248
Using BI Server Scheduler 249
Scheduling a Query 249
Retrieving Scheduled Results 250

Appendix A: Accessibility and Technical Support 251

Accessibility 253
Microsoft Accessibility Options 254
Technical Support 254

Index 257

Chapter 1

Introducing BI Query

About BI Query **3**
BI Query Admin, User, and Update 3

Basic Concepts **4**
Database Components 4
Data Model Components 4
Queries 7
Reports 8
Types of Data Models 8
BI Query Operation Modes 11
Using BI Query with BI Server 12

About BI Query

Hummingbird BI Query is a query and reporting tool that provides a comprehensive solution for accessing, analyzing, and presenting data stored in enterprise databases. BI Query lets users extract the information they need using a data model—a graphical representation of the database. Using BI Query's graphical approach, users can form queries without needing to know SQL (Structured Query Language—the language used for retrieving data from most databases).

For Administrators BI Query provides the flexibility to tailor information access to the exact needs of business users. The administrator makes business-critical information available while maintaining data security, quality, and integrity.

For Business Users BI Query provides an easy-to-use, visual way to query databases, integrate data with other applications, and generate reports.

BI Query Admin, User, and Update

The BI Query product line consists of three applications—Admin, User, and Update:

BI Query Admin Lets the BI Query administrator manage the use of the program by users. The BI Query administrator can design data models, set permissions for users, set passwords, and control access to the database and the functionality of BI Query. With permission from the DBMS administrator, the BI Query administrator can also update the tables in the database.

BI Query User Lets users run queries provided by others and, depending on their user permissions, create ad hoc queries of their own.

BI Query Update Provides users with the same functionality as BI Query User with the additional ability—depending on their DBMS permissions—to update tables in the database.

Basic Concepts

The following sections describe the basic concepts behind BI Query's graphical approach to extracting information from corporate databases.

Database Components

A database is a collection of related information. The basic components of a database are as follows:

Tables In a relational database, information is held in tables. A table usually relates to something in the real world. For example, a database might store customer names and addresses in one table, products in another, stock levels in another, purchase orders in another, and so on.

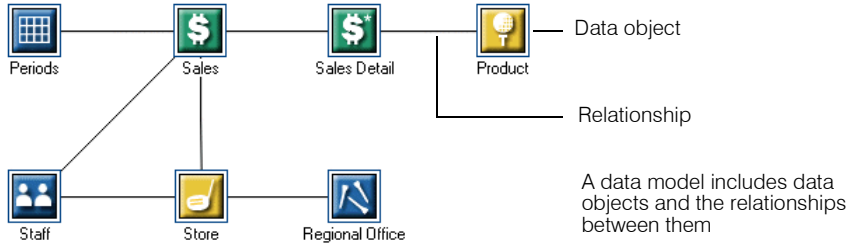
Columns Tables are made up of columns and rows. Each column represents an attribute of the object that the table represents. For example, customers have names, addresses, fax numbers, and so on. A table for customer information would have a column for each of these attributes. (Columns are also known as attributes.)

Rows Each row in a table is an instance of the object—that is, each row in the customer table gives us all the information about a customer—specific name, specific address, and so on. (Rows are also known as records.)

DBMSs The collection of programs that manage a database constitute a database management system (DBMS). A DBMS lets users examine and manipulate data in "real world" terms—customers, orders, products—without needing to know how the computer actually stores the information.

Data Model Components

A data model is a graphical representation of the data in a database.

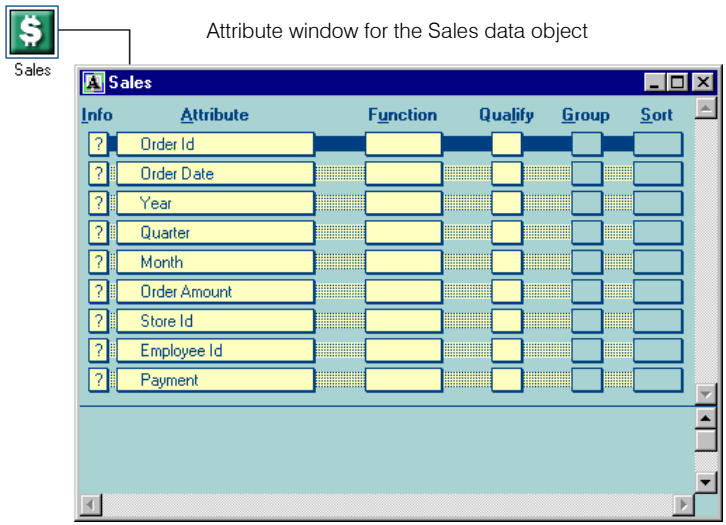


When you use BI Query to get information from a database, you work with a data model. Depending on the BI Query application you are using and the permissions assigned to you, you may be able to customize data models to suit your needs. Administrators, who use BI Query Admin to design corporate data models, set permissions and preferences for each data model. Users can change the preference settings, but not the permissions.

Data models consist of the following components:

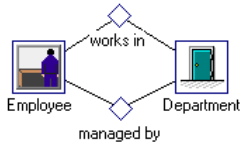
Data Objects Data objects are rectangular or graphical icons that represent the tables stored in the database. Each data object contains one or more attributes.

Attributes An attribute represents a column of data in a database table. When you double-click a data object in a data model, an attribute window opens, listing the attributes stored in the data object.



Relationships A relationship connects two data objects together and indicates that the connected objects contain at least one attribute in common. For example, an Employee data object might be related to a Department data object on the basis of a common Manager attribute. In order to include the attributes from two or more data objects in a query, the data objects must be related.

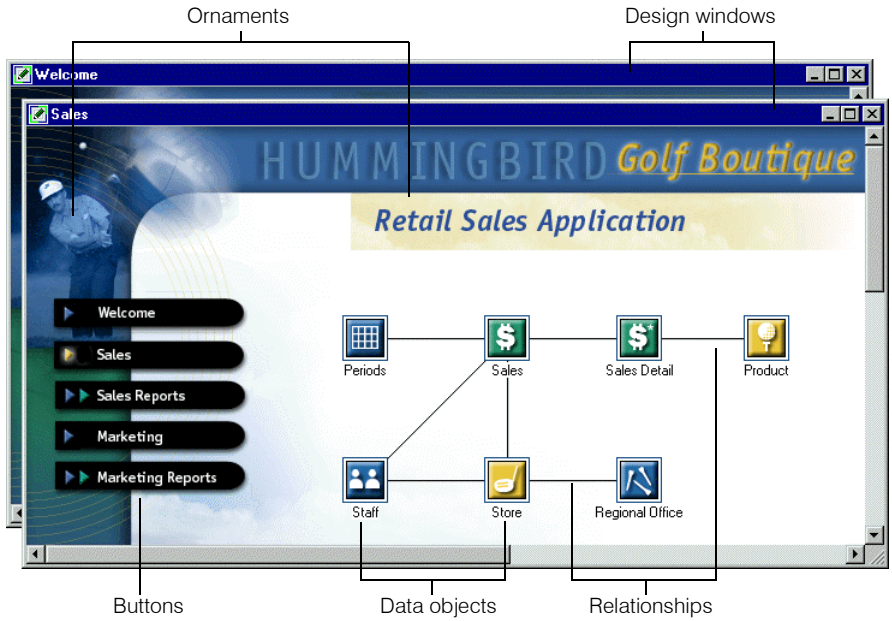
Data objects can have more than one relationship, so that you can get different information using the same objects. BI Query represents relationships as connecting lines between objects. Relationships can also appear with a diamond icon and a name.



You could use the *works in* relationship to find information about an employee who works in a particular department

You could use the *managed by* relationship to find information about an employee who manages a particular department

Design Windows Design windows are the workspace in which administrators design data models and users formulate queries. Design windows contain the data objects that represent tables in the database and the relationships that tie them together. Design windows can also contain buttons (for navigating between design windows and automating tasks) and ornaments (such as graphics and text) that can provide information and improve the usability of the model.



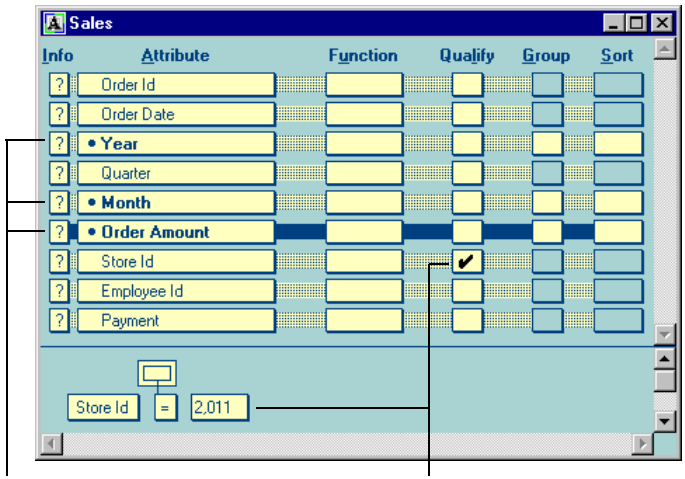
Buttons Buttons automate frequent activities. They let users connect automatically to the database, submit multiple queries, combine the results, and generate a report—all with a single click of the mouse. Buttons can also display design windows, save results to files, export results to other applications, open associated document files, and launch other applications such as Visual Basic and Excel.

Ornaments Ornaments are text and graphical objects such as titles, logos, borders, backgrounds, and notes that provide additional information, act as visual organizers, and improve the appearance of design windows.

Queries

A query is a request for information from a database. In order to retrieve information using the data objects and relationships in a data model, you create a query and submit it to the database. The results returned by the database for a given query are known as a results set.

The first step in creating a query is to select attributes from at least one data object. You may also want to qualify one or more attributes to restrict the results to the particular information that interests you—such as the sales information for a particular store.



This query returns all year, month, and order amount data for all stores with ID 2011

Selected attributes specify what data to retrieve from the database

Qualified attributes filter the data

The designer of a data model typically creates queries and saves them with the data model; users can then open the queries from the data model and submit them to the database. The designer can also let users edit existing queries and create their own (called ad hoc queries).

Reports

Once you've gathered the data you want by querying, you may need to present it in a report. BI Query has two report generators, BI Query Reports and BI Query Standard Reports, that let you produce your own professional looking reports.

For more information on BI Query Reports, see the *BI Query Reports User's Guide*.

BI Query Reports provides a flexible reporting environment that includes tables, crosstabs, charts, and maps. You can open BI Query Reports from within BI Query or directly from the Hummingbird BI program group in the Windows Start menu. BI Query Reports has its own online Help system, which you can access from its Help menu.

BI Query Standard Reports is an integrated component of BI Query. For more information on Standard Reports, see "About Standard Reports" on page 151.

Types of Data Models

BI Query provides two types of data models: split and combined.

Split Data Models

A split data model lets users add their own customizations to a data model (including changes to their preferences and other default settings) and retain those customizations when a new version of the data model is distributed. This type of data model is suitable for most sites.

A split data model is stored as two layers—an administrator layer and a user layer.

Administrator Layer The administrator layer of a split data model is the data model that the administrator creates and edits. It consists of a file with the extension `.gqa`, a `Queries` folder, a `DataVals` folder, and a `Reports` folder. The administrator-layer file includes all the components that make up a basic data model. The administrator distributes this file, plus the associated folders, to all users.

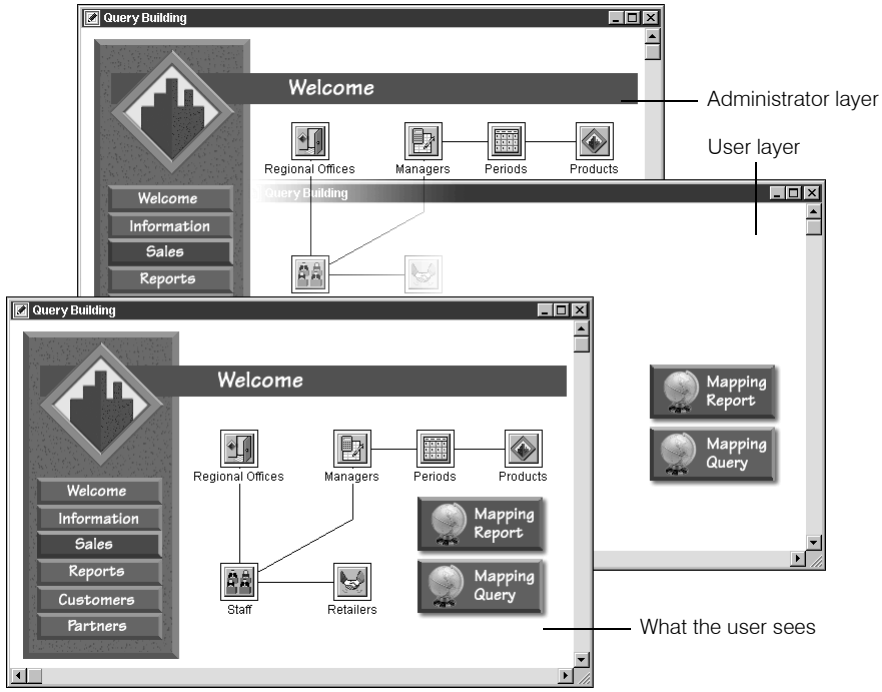
User Layer The user layer of a split data model consists of a file with the extension `.gqu`, a `Queries` folder, a `DataVals` folder, and a `Reports` folder. Along with the administrator layer, the administrator distributes an essentially empty copy of the user-layer file to all users. The user-layer file contains a few default settings, such as preferences. It also contains the path to the administrator layer; when a user opens the user-layer file, BI Query uses the path to find the administrator layer. BI Query then combines all the elements stored in the user-layer file with the elements stored in the administrator-layer file.

The following table shows the distribution of components in a split data model:

Layer Component	Administrator Layer	User Layer
Data model file	*datamodel*`.gqa`: • design windows • data objects • relationships • calculated attributes • buttons • ornaments • drawing objects • permissions • default preferences	*datamodel*`.gqu`: • design windows • calculated attributes • buttons • ornaments • drawing objects • preferences
Queries folder	• query files • prompts file • variables • standard report specifications	• query files • prompts file • variables • standard report specifications
Reports folder	• BI Query Reports files	• BI Query Reports files
Data Values folder	• data values query files • data values results files • data sets	• data values query files • data values results files • data sets

The first time a user opens a split data model, the user sees only the data model stored in the administrator layer. When the user layer is stored in a folder separate from the administrator layer, BI Query creates empty `Queries`, `DataVals` and `Reports` folders to store user-defined queries, prompts, variables, data values files, data sets, and reports. BI Query saves any other customizations the user adds (such as extra buttons, ornaments, and design windows) in the user-layer file.

The next time the user reopens the data model, the user sees the administrator's data model as well as all the customizations he or she has added. The changes made by one user don't affect those made by other users. Users can't make changes to the administrator layer.



Combined Data Models

A combined data model consists of one data model file (with extension `.gql`) and the associated `Queries`, `DataVals`, and `Reports` folders. Each time the administrator revises a combined data model and distributes it to users, any user customizations are overwritten.

Typically, combined data models are created for organizations that don't allow users to make any changes to the data model. The following table lists the components of a combined data model:

Layer Component	Combined Data Model
Data model file	*datamodel*.gql: • design windows • data objects • relationships • calculated attributes • buttons • ornaments • drawing objects • permissions • default preferences
Queries folder	• query files • prompts file • variables • standard report specifications
Reports folder	• BI Query Reports files
Data Values folder	• data values query files • data values results files • data sets

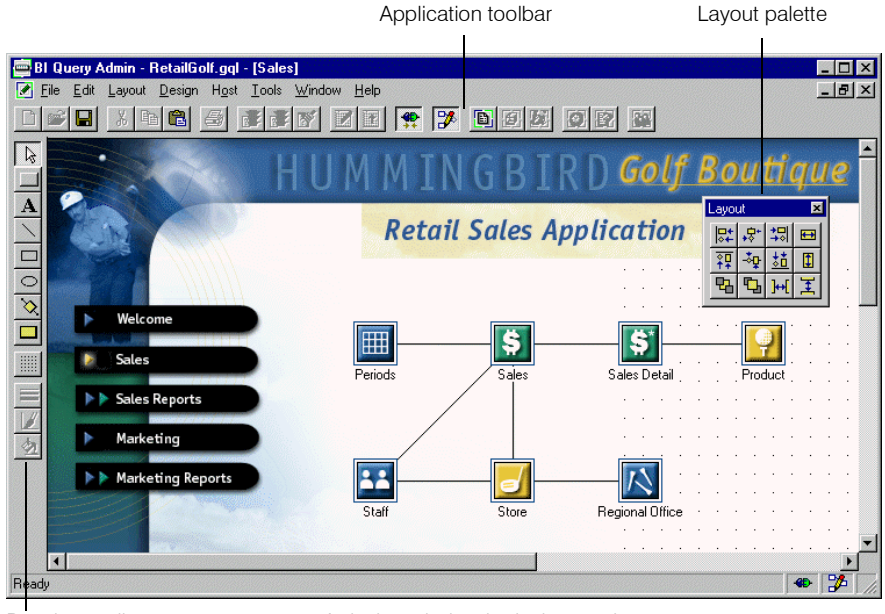
BI Query Operation Modes

BI Query operates in one of two modes depending on the type of activities you want to perform.

Run Mode Lets you perform activities that require access to the database (submitting queries, updating tables, submitting SQL scripts, and so on).

Design Mode Lets you modify the design and layout of the data model, create buttons and ornaments, and perform other activities that do not require a connection to the database.

You can switch between these two modes by clicking the Design Mode button on the Application toolbar.



The Drawing toolbar and Layout palette are available only in Design Mode. The tools on the Drawing toolbar let you create objects and assign properties to them (color, line thickness, and so on). The Layout palette provides tools for resizing and aligning objects.

Using BI Query with BI Server

BI Server is an application server that uses a common set of services and a common repository of information to provide data for Hummingbird BI applications. The BI Server Repository is a storehouse for enterprise information produced using Hummingbird BI applications. It stores data models, queries, results sets, reports, multi-dimensional data sources, and associated metadata. If you are in a BI Server environment, you will be able to take advantage of BI Server's publishing, retrieving, scheduling, and security features.

Specific terms are used to distinguish Server-related actions from local (computer-based) actions. Locally saved data (on your desktop computer) is *saved* or *opened*. Material stored on the BI Server Repository is *published* or *retrieved*.

If you've been assigned the appropriate system permissions, you can publish BI Query information to the Repository and you can retrieve information you've published as well as information published by others. By publishing to the Repository, you ensure that the information is accessible to other BI Query users and to BI Web users. Because you can set security on items you publish, publishing also provides a secure way to share your information with other users.

For more information on using BI Query in a BI Server environment, see "Working with BI Server" on page 239.

Chapter 2

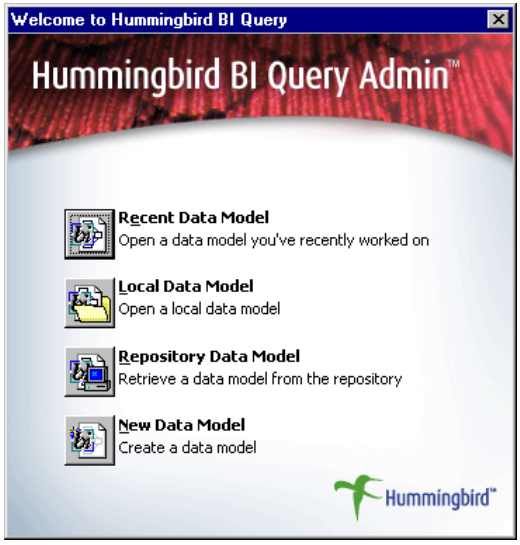
Getting Started

Starting BI Query **17**
Opening Data Models 18
Retrieving Data Models 19
Finding the Administrator Layer in a Split Data Model 20
Working Online or Offline 21

Connecting to a DBMS **21**
Disconnecting from the DBMS 23

Starting BI Query

When you start BI Query, the Welcome to Hummingbird BI Query dialog box automatically prompts you to open a data model. (You can also open a data model after you have started the program.)


Welcome to Hummingbird BI Query
Hummingbird BI Query Admin™
Recent Data Model
Open a data model you've recently worked on
Local Data Model
Open a local data model
Repository Data Model
Retrieve a data model from the repository
New Data Model
Create a data model
Hummingbird™

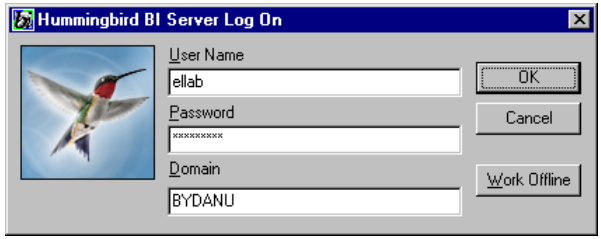
The types of data models you can open at start-up are as follows:

- Recent data models are those data models on your system that you have recently accessed.
- Local data models are all the data models on your system; you do not need to log on to BI Server to access these models.
- Repository data models are the data models available in the BI Server Repository. Before you can access a Repository model, you must log on to BI Server.
- New data models are empty models that you can create and develop. Only BI Query administrators can open new data models.

For more information on opening new data models, see the *BI Query Data Models User's Guide*.

The BI Server Environment

Before you are prompted to select a data model, the BI Server Log On dialog box may open. This dialog box indicates that your corporate querying environment includes a central storehouse for data models, queries, results, reports, and data sources (the BI Server Repository).



You can log on to use the Repository, or you can work offline. If you log on, you can publish, secure, and retrieve queries and reports in the Repository. You can also schedule queries and reports. If you choose to work offline, you do not have access to the Repository. However, you can still connect to the database, submit queries, and work with query results.

Opening Data Models

A data model can be either split or combined. Combined data model files have the extension `.gql`. To open a split data model, open the user layer (extension `.gqu`).

If you're working online and you open a local copy of a published data model, you'll receive a message if a version of that model has been published more recently than the date of the local copy. Depending upon the settings specified when the data model was published, you may be able to choose whether to retrieve the more recent version, or you may be required to do so.

To start BI Query and open a data model:

1 On the Start menu, navigate to the Hummingbird BI program group, point to Hummingbird BI Query, then click your BI Query application (one of BI Query Admin, BI Query Update, or BI Query User).

2 If the BI Server Log On dialog box appears, do one of the following:

- To log onto the Repository, type your user name and password. If the Domain box appears and is enabled, type your domain name. (If you're unsure what information to provide in the dialog box, check with your BI Server administrator.) Click OK.
- To work without access to the Repository, click Work Offline.

3 In the Welcome to Hummingbird BI Query dialog box, do one of the following:

- To open a data model that you have used recently, click Recent Data Model. In the Recent Data Models dialog box, select the data model you wish to open, then click Open.
- To open a local data model click Local Data Model. In the Open Data Model dialog box, select the data model file you want, then click OK.
- To open a new data model, click New.

4 If the Enter Data Model Password dialog box appears, type your password. If a password has not been set, this dialog box does not appear.

5 A message box may open to indicate that a newer version of the data model exists in the database. To open the newer version, click Yes.

To open a data model from within BI Query:

1 On the File menu, click Open. In the Open Data Model dialog box, select the data model file you want, then click OK.

2 If the Enter Data Model Password dialog box appears, type your password.

3 A message box may open to indicate that a newer version of the data model exists in the database. To open the newer version, click Yes.

Retrieving Data Models

When you retrieve a data model from the BI Server Repository, BI Query copies it locally. Any queries, data values files, data sets, connection files, standard reports specifications, and BI Query Reports files that were published with the data model are retrieved and stored in the appropriate folders. You must be logged onto BI Server to retrieve data models.

For more information, see the *BI Query Data Models User's Guide*.

If you have the appropriate BI Server and BI Query permissions, you can make and save changes to the copy, then publish it to the Repository again. (The source data model remains on your local machine as originally saved unless you delete it.)

To retrieve a data model:

1 Do one of the following:

- If you're starting BI Query, in the Welcome to Hummingbird BI Query dialog box, click Repository Data Model.
- If BI Query is already running, close the data model you're using, then on the File menu, click Retrieve.

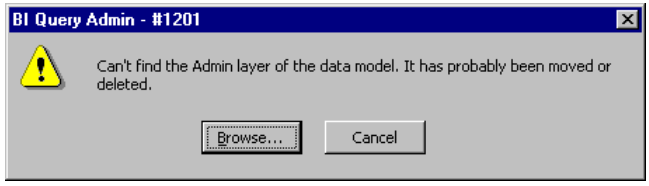
2 In the Retrieve Data Model dialog box, select the data model you want, then click OK.

Finding the Administrator Layer in a Split Data Model

Each user layer of a split data model contains an internal link to the administrator layer. If the administrator-layer file has been moved or renamed, BI Query will not be able to find it when you open the model. In this case, BI Query prompts you for the new location. To reset the internal link, you must locate the file, then save the data model.

To find the administrator layer when it's been moved or renamed:

1 In the alert box that opens when you try to open the data model, click Browse.



2 In the Open Data Model dialog box, find and open the administrator-layer file.

3 On the File menu, click Save.

Working Online or Offline

If you are in a BI Server environment, you have the option of working online (logging on to the BI Server Repository) or offline. You can log on to BI Server when you start up BI Query. You can also log on after you have started BI Query.

To log on to BI Server from within BI Query:

1 On the File menu, click Work Online.

2 In the BI Server Log On dialog box, type your user name and password. If the Domain box appears and is enabled, type your domain name into it. Click OK.

To work offline after you have logged on to BI Server:

On the File menu, click Work Online.

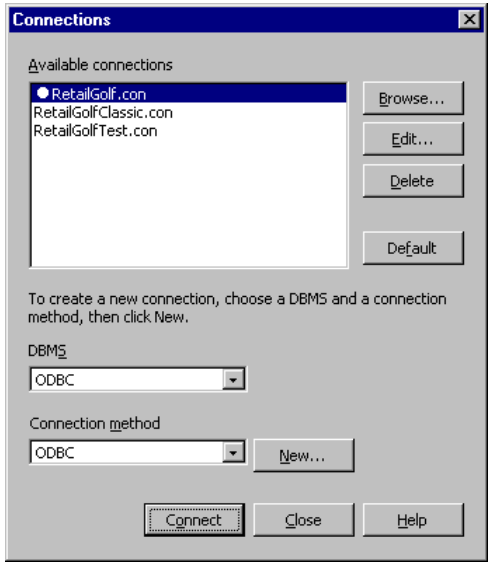
Connecting to a DBMS

For information on creating connection files, see the *BI Query Data Models User's Guide*.

In order to query a database, BI Query must connect to the corresponding database management system (DBMS) running on the server. To make the connection to a DBMS, BI Query uses a connection file. Each connection file stores the information about the DBMS and the middleware that is necessary to make the connection.

Typically, the BI Query administrator includes at least one connection file in the data model (the default); the administrator may configure the model so that it automatically connects to the database each time you open the model or submit a query. Even if the model has not been configured to connect automatically, BI Query attempts to connect if you submit a query and a connection file is available.

You can connect a data model to the corresponding DBMS by using a default connection file or by specifying a different one. To specify a particular connection file, use the Connections dialog box.



You need to specify a connection file if the data model does not have a default. You can also create your own connection files.

To connect using the default connection file for the data model:

1 On the Host menu, click Connect.

2 If the Enter Connection Information dialog box appears, specify the requested information (such as database user name and password). Click OK.

To connect using a different connection file:

1 On the Host menu, click Connections. The Connections dialog box opens.

2 From the Available Connections list, select a connection file. If the file you want does not appear in the Available Connections list, click Browse and locate the desired file on your system.

3 Click Connect.

4 If the Enter Connection Information dialog box appears, type the requested information, then click OK.

Disconnecting from the DBMS

BI Query automatically disconnects from the DBMS whenever you close a data model or exit from BI Query. You may wish to disconnect at other times. You can continue to carry out many activities in BI Query while not connected, but you cannot submit queries, send an SQL script to the DBMS, or update information in the database.

To disconnect from the DBMS, do one of the following:

- On the Host menu, click Disconnect.
- On the Application toolbar, click the Disconnect button.

Chapter 3

Building Queries

Query Basics **27**
Building Queries 28
Submitting and Clearing Queries 29
Cancelling Queries 30
Reordering Columns Before Submitting a Query 31
Sorting Rows 31
Viewing Query Specifications 33
Saving Queries 33
Opening Saved Queries 35

Qualifying Queries **36**
Using the Qualification Tree 36
Adding, Modifying, or Deleting Qualifications 38
Qualifying with Available Data Values 39
Applying Multiple Qualifications with And/Or 41
Combining Qualifications 42
Using Negative Qualifications 42
Using Query Modifiers 43
Qualifying One Attribute with Another 44

Using Qualification Operators **45**
Matching Character Patterns 46
Using Wildcard Characters 46
Making Comparisons 47
Matching Items in a List 47
Finding a Range of Values 48
Matching Any/Matching All 48
Determining if Data is Missing 49
Applying Your Own Operators 50

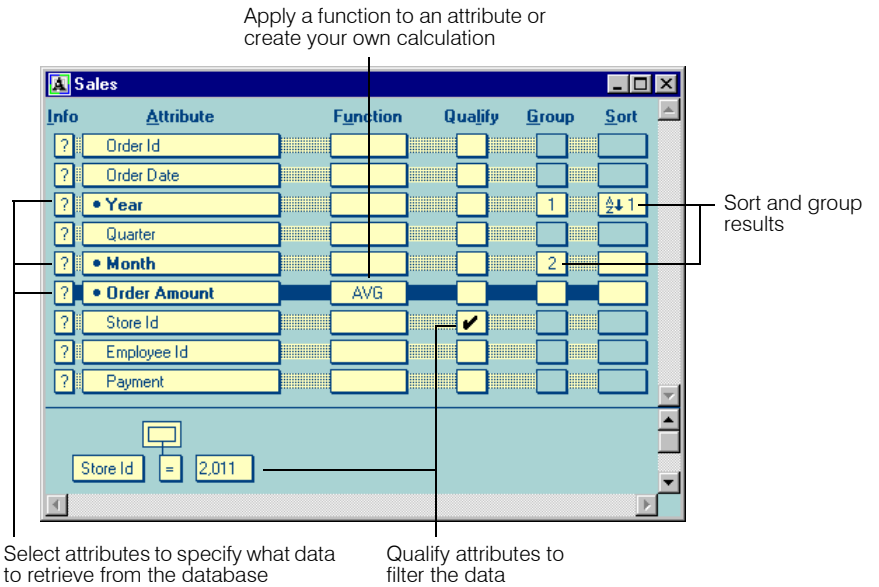
Querying Multiple Data Objects **50**
Removing Unnecessary Joins from Queries 51

Updating Queries **52**
Refreshing a Query 52
Using Legacy Joins As Is 53

Query Basics

In order to retrieve information from the database, you need to build queries (sometimes called ad hoc queries) that ask the questions you want answered. To build a query, you choose categories of information and then decide how much of the information you actually want to receive in the results. You may want the query to perform calculations on the data it retrieves before providing you with results. You may also want to specify how rows are to be sorted and how columns are to be ordered in the results.

You build a query in the attribute window.



When you submit a query, you choose whether to send the results directly to the results window for use in BI Query, or to a text file for use in other applications, or to a new table in the host database for use in creating other queries.

For more information, see "Using BI Server Scheduler" on page 249.

You can also schedule queries to run at certain times if you have access to the BI Server Repository and have the appropriate publishing and scheduling permissions.

Building Queries

You build a query by selecting attributes in a data object. The selected attributes appear as column headings when your query brings back the information. As soon as you select one attribute, you have built a query. It is up to you to decide how much more you want your query to include.

For example, if you select the Store Name attribute in a Store data object and then submit the query to the database, it brings back the names of all the stores in the database table, in a column labeled Store Name. If you also select the City attribute in the same data object and submit the query, it brings back the store names, in a column labeled Store Name, and the city in which each store is located, in another column labeled City.

For more information on combining query results, see "Combining Results Sets" on page 140.

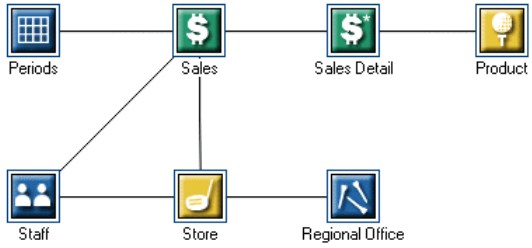
You can select as many attributes as you like in each data object. And you can select attributes from as many data objects as you like within one data model window. (If your data model has data objects in more than one window, you can build separate queries in each window, then use one of several methods for combining them.)

If your data model contains many data objects, you can find the ones you want using the Find and Replace dialog box. For more information on searching a data model, see the *BI Query Data Models User's Guide*.

To select attributes:

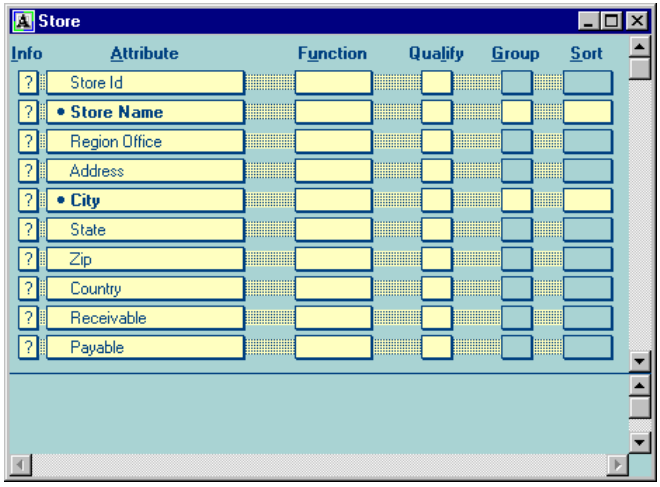
1 If a query is already open and you want to build a new one, click New on the Query menu.

2 Double-click the data object for which you want to retrieve information.



Double-click the Store data object to open its attribute window

3 In the attribute window, click the attributes that represent the categories of information you want to retrieve.



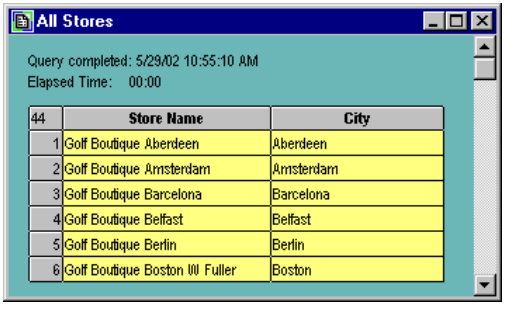
For more information, see "Querying Multiple Data Objects" on page 50.

4 If an additional data object is connected by a relationship with the first, you can select attributes for your query from that data object as well.

5 If your query is complete, click Run on the Query menu, or, on the Application toolbar, click the Run Query button.

Submitting and Clearing Queries

Once you've built a query, you can submit it to the database. The results of the query are displayed in a separate results window named after the query.



The results window for the All Stores query

You can build your entire query and then submit it. Or you can specify some initial qualifications, submit the query, review the results, and then revise the query to retrieve exactly the set of results you want.

To submit a query:

1 Do one of the following:

- Build a new query.
- Open a previously saved query: on the Query menu, click Open and specify a query in the Open Query dialog box.
- Retrieve an existing query from the BI Server Repository. For more information, see "Retrieving Queries" on page 241.

2 On the Query menu, click Run or, on the Application toolbar, click the Run Query button.

Queries remain current even after they have run. This may not be obvious if the query is not on the screen. You should always clear the previous query before starting a new one (unless you want to modify the previous query).

To clear a query, do one of the following:

- On the Application toolbar, click the New Query button.
- On the Query menu, click New.

Cancelling Queries

After submitting a query, you may realize that you need to make some changes, or you may decide that it is taking too long to run. You can cancel a query only after rows have been returned. Not all databases support this function.

To cancel a query, do one of the following:

- On the Query menu, click Cancel.

- On the Application toolbar, click the Cancel Query button.

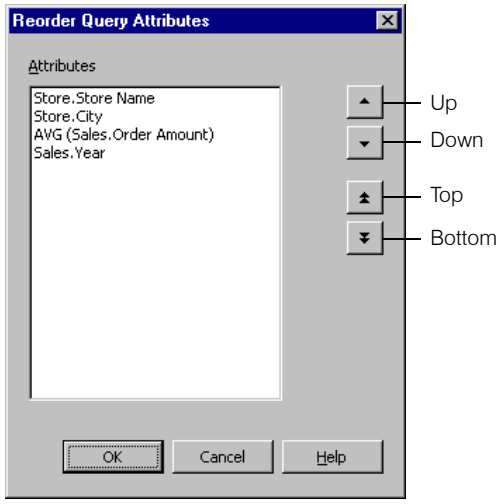
Reordering Columns Before Submitting a Query

For more information about reordering columns in BI Query Reports, see BI Query Reports Help.

Columns appear in results windows and reports in the order in which you select attributes while creating the query. You can change that order before submitting the query, after you have results, or in a report.

To specify the column order before submitting the query:

1 On the Query menu, click Reorder Attributes. The Reorder Query Attribute dialog box opens. The top-to-bottom order of attributes in the Attributes list corresponds to the left-to-right order of columns in the results window or report.



2 From the Attributes list, select the attribute whose position in the list you want to change. Use the Up, Down, Top, or Bottom buttons to move the selected attribute to the desired position.

3 Click OK.

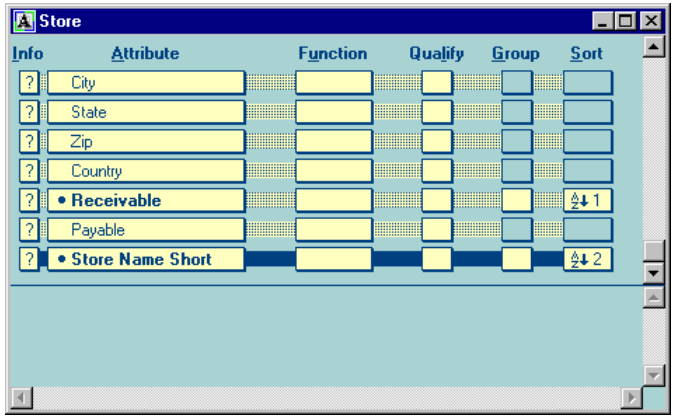
Sorting Rows

When you submit a query, BI Query displays the rows of results as the DBMS returns them from the database, unless you specify a different way of sorting them. You can sort rows in ascending or descending alphabetical, numerical, or chronological order, as appropriate.

You can also sort results after you've submitted a query. For more information, see "Getting Results in Sorted Order" on page 136.

Use the Sort boxes in the attribute window to sort. BI Query sorts results first by the data returned for the attribute with 1 in its Sort box, then by the data returned for the attribute with 2 in its Sort box, and so on. You can sort attributes from more than one attribute window in the same query.

For example, suppose that you select the Store Name Short and Receivable attributes in a Store data object, then click the Sort boxes for Receivable and Store Name Short, in that order. When you submit the query, BI Query sorts the results numerically by receivable amounts first, then alphabetically by abbreviated store name.



To specify a sort order in an attribute window:

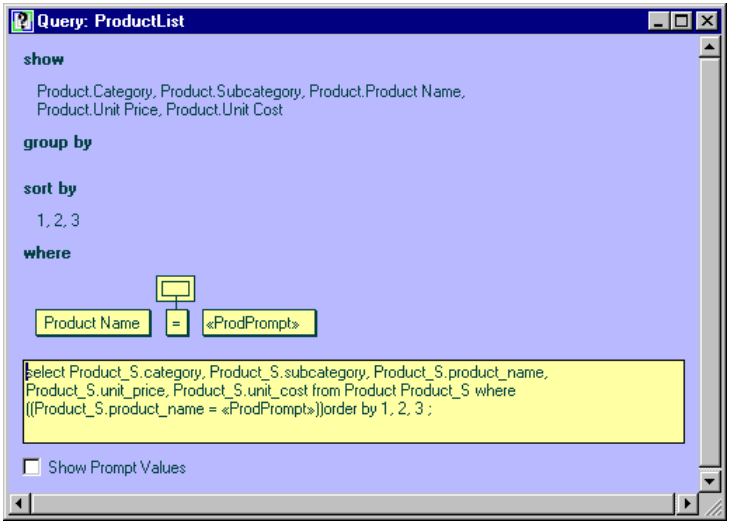
1. In an attribute window, select the attributes you want to sort.
2. Click the Sort box for each attribute in the sort order you want.
3. To reverse the sort method (ascending or descending) for any sorted attribute, click the Sort box again for that attribute.

Ascending sort

Descending sort

Viewing Query Specifications

You can view query specifications (selected attributes, functions, sort order, and group order for each data object) in the Query window.



For more information, see "Modifying the SQL for a Query" on page 207.

If the Show SQL String preference is set, the Query window displays the corresponding SQL string. If the Edit Queries permission is assigned, you can modify the SQL string directly in the Query window.

To view query specifications:

1 Open or create the query.

2 On the Query menu, click Show.

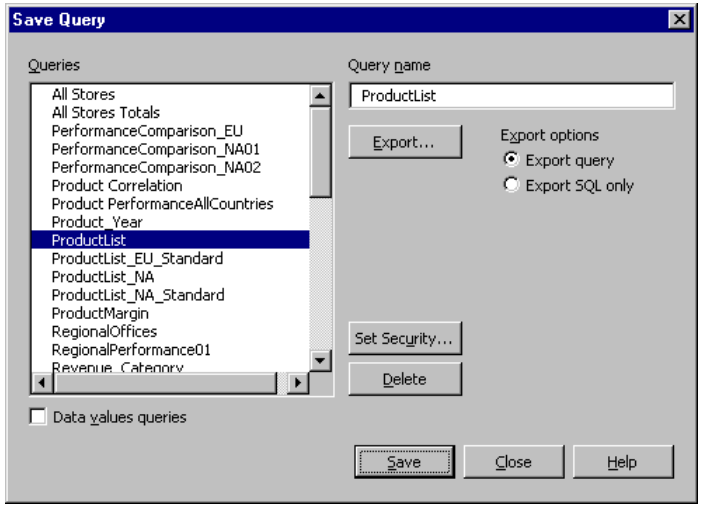
Saving Queries

For information on deleting saved queries, see the Help for your BI Query application.

If the Save Queries permission is assigned, you can save queries in order to use them again in BI Query. You can also export the underlying SQL text for use in other applications. You can save a query before or after submitting it to the DBMS. You can open saved queries in BI Query, then run or delete them.

To save a query:

1 On the Query menu, click Save. The Save Query dialog box opens.



2 In the Query Name box, type a name for the query.

3 Do one of the following:

- To save the query as a file with extension .qry, click Save.
- To save the query as a data values query file in the DataVals folder, click Data Values Queries, then click Save. For more information, see "About Data Values Files" on page 79.
- To save the query in a location other than the Queries or DataVals folders, click Export. In the Export Query dialog box, select a location for the file, then click Save.

Note: If you save a query in the results or report window, BI Query saves the query that generated the results or report displayed. If you are in any other window, BI Query saves the current query.

Opening Saved Queries

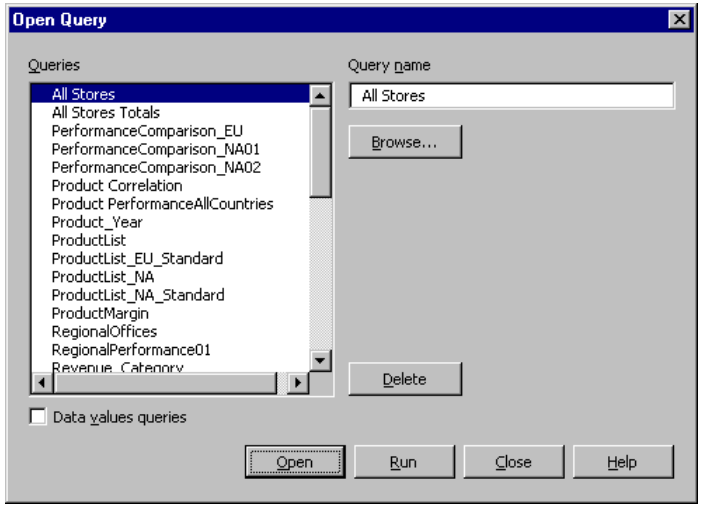
For information on retrieving a published query, see "Retrieving Queries" on page 241.

You can submit or modify a saved query. Locally stored queries (in a data model that you can access without logging on to BI Server) are *opened*; queries stored in the BI Server Repository must be *retrieved*.

To open a saved query:

1 On the Query menu, click Open.

2 In the Open Query dialog box, select the query you want to open.



3 Do one of the following:

- Click Run to open the query and submit it to the DBMS.
- Click Open to open the query without submitting it.

The Query window opens, displaying both BI Query and SQL information about the query. While the opened query is current, you can alter it by selecting or deselecting attributes. When you have finished editing the query, save and submit it.

Qualifying Queries

You can limit query results to a subset of the available information by qualifying one or more attributes. In general, a qualification takes the following form:

```
attribute operator testvalue
```

where

- *attribute* is the name of the attribute you are qualifying
- *testvalue* is the value or values you want to compare against the values of the attribute
- *operator* is the logical operator that specifies the nature of the comparison

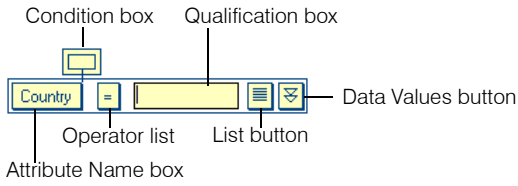
For example, if you want to return rows that contain data for sales greater than $10,000, you could qualify the Sales attribute using the following expression:

```
Sales > 10000
```

You qualify an attribute by clicking its Qualify box in the attribute window, then specifying the qualification in the qualification tree.

Using the Qualification Tree

The qualification tree appears at the bottom of the attribute window when you click the Qualify box for an attribute.



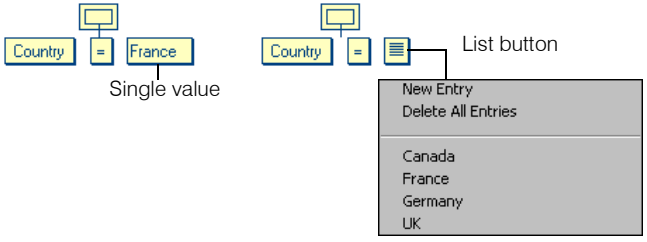
Attribute Name Box Displays the name of the qualified attribute.

Condition Box Lets you combine multiple qualifications. You can also use the Condition box to reverse the effect of a particular qualification. To remove a qualification from a query, click the Condition box, then hit the Delete key.

Operator List Lets you specify the type of restriction you want. For example, you can apply the Greater Than (>) operator to find all rows in which the specified attribute has a value greater than the specified comparison value.

If your database is case-sensitive and the Case-sensitive preference is set, when you type a qualification, the characters must be in the case in which they are stored in the database. If this preference isn't set, case isn't important.

Qualification Box Lets you specify the comparison value for the attribute. For example, by qualifying the Country attribute with France, Germany, and Spain, you can restrict the results to those countries. After entering a qualification, click outside the qualification tree. If you entered one value, it appears in the qualification tree; if you entered more than one value, click the List button to view the values.



List Button Lets you add, modify, or delete entries in the list of qualification values. Values that you enter in the Qualification box are displayed in a list when you click the List button. By clicking a value in the list, you can add that value to the Qualification box. Alternatively, use the Up or Down arrow keys on your keyboard to display each value in the Qualification box.

Data Values Button Lets you insert into the qualification one or more of the following:

- Data values. For more information, see "Qualifying with Available Data Values" on page 39.
- Data sets. For more information, see "Using a Data Set to Qualify an Attribute" on page 194.

- Prompts. For more information, see "Inserting a Prompt in a Query" on page 102.
- Variables. For more information, see "Using Variables to Qualify an Attribute" on page 119.
- Subqueries. For more information, see "Creating and Inserting Subqueries" on page 191.

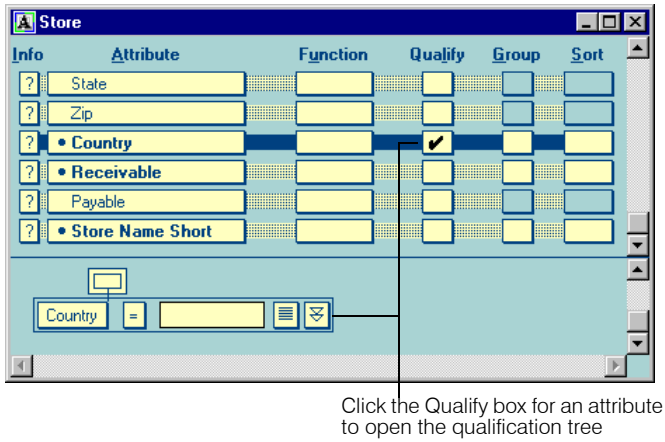
Adding, Modifying, or Deleting Qualifications

To cancel a qualification, click the Condition box (if there are no values in the Qualification box). If there are qualification values, click the Condition box, then press the Delete key.

You can qualify multiple attributes in a given query and qualify a given attribute more than once. You do not need to select an attribute to qualify it. Selecting an attribute indicates only that the attribute appears in the results.

To qualify an attribute:

1 In the attribute window, click the Qualify box for an attribute you want to qualify. A qualification tree appears at the bottom of the attribute window.



Click the Qualify box for an attribute to open the qualification tree

For more information on available operators, see "Using Qualification Operators" on page 45.

2 In the qualification tree, select an operator from the Operator drop-down list. (The default operator is =.)

3 Qualify the attribute by doing one of the following:

- In the Qualification box, type the qualification value. To type more than one value, hit the down arrow key between successive values.

- Click the List button and select a previously entered value from the pop-up menu that opens.
- Click the Data Values button and select one or more data values, data sets, prompts, variables, or subqueries from the pop-up menu that opens.

4 Press Enter (or click outside the qualification tree).

5 On the Query menu, click Run.

To add a new comparison value to the qualification:

1 Click the List button, then click New Entry from the pop-up menu that opens.

2 In the Qualification box, type the new value, then press Enter (or click outside the qualification tree).

To modify existing values:

1 If you have qualified with multiple values, click the List button, then click the value you want to modify.

2 Click in the Qualification box, then type over the value to modify it.

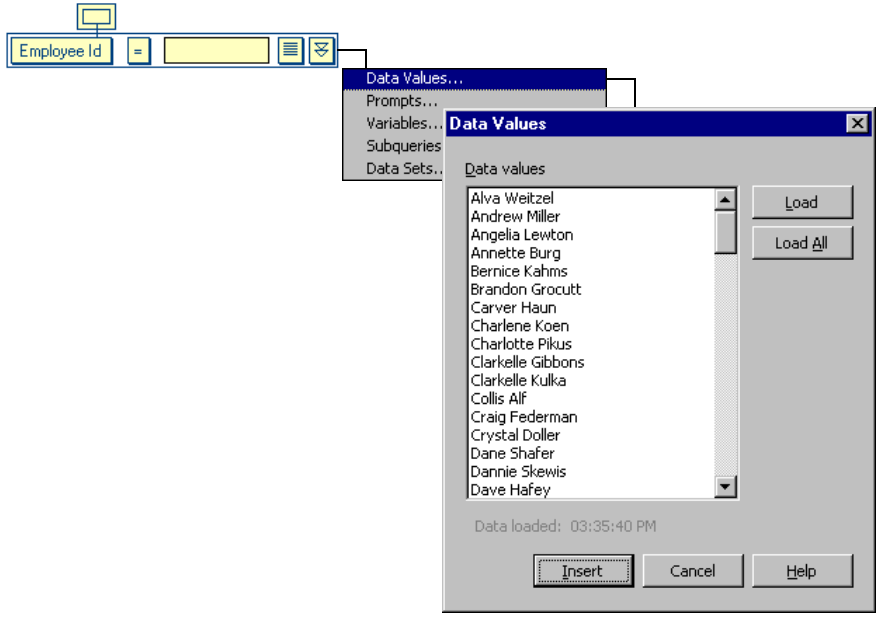
To delete values, do one of the following:

- Click in the Qualification box to highlight the value, then press the Delete key.
- If you have qualified with multiple values, click the List button. To delete the entire set of values at once, click Delete All Entries. Otherwise, click the value you want to delete, then, with the value highlighted in the Qualification box, press the Delete key.

Qualifying with Available Data Values

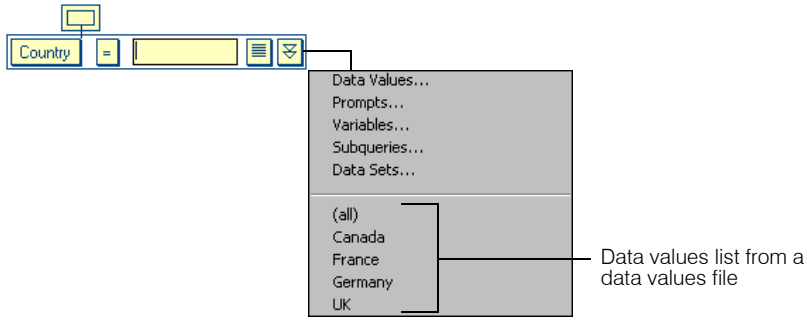
The Data Values button in the qualification tree displays a pop-up menu that lets you select a data value (or set of data values) to insert into the qualification for an attribute.

When you click Data Values from the pop-up menu, BI Query submits a default `SELECT DISTINCT` query that retrieves all the data values in the database for that attribute and displays them in the Data Values dialog box, unless the BI Query administrator has disabled data values for that attribute.



For more information on data values files, see "About Data Values Files" on page 79.

If a data values file has been created for the attribute, BI Query instead displays the list from that file. The list can be a timesaving subset of the data values stored in the database.



If you need to access all the data values in the database instead of just those in the subset, you can click Load All in the Data Values dialog box.

To qualify an attribute with a data value:

1 In the attribute window, click the Qualify box for the attribute you want to qualify.

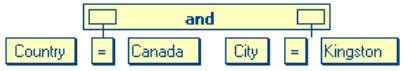
2 In the qualification tree for the attribute, click the Data Values button. From the pop-up menu that opens, do one of the following:

- Click the value from the list of available values at the bottom of the menu. These values appear only if the attribute has a data values file associated with it.
- Click (all) to insert all listed values into the qualification.
- Click Data Values to open the Data Values dialog box. Use this dialog box to select the value or values that you want to insert into the qualification, then click Insert.

Note: You can view the data values for *any* attribute (and insert them into the qualification for any attribute) using the Choose Data Values dialog box: on the Edit menu, click Data Values.

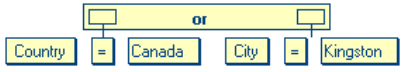
Applying Multiple Qualifications with And/Or

When you qualify more than one attribute, or when you qualify the same attribute more than once, BI Query assumes that the query must satisfy both qualifications. This is reflected in the qualification tree, where the word and appears between two qualifications. You can change and to or.



To specify that the query must satisfy either or both of the qualifications:

Click the word and. It changes to or.

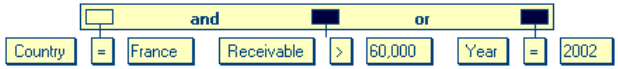


Combining Qualifications

You can avoid ambiguities in a query by combining two or more qualifications so that they take precedence over those that are not combined.

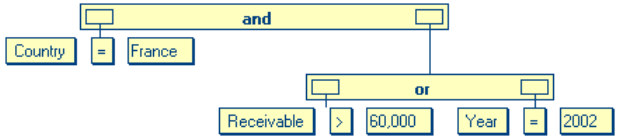
To combine qualifications:

1 In the qualification tree, select the qualifications to be combined by holding down the Shift key and clicking the corresponding condition boxes.



2 On the Query menu, click Qualification, then click Combine.

3 To change the condition in the new branch from and to or, click the word and.



To separate combined qualifications:

1 In the qualification tree, click the condition box corresponding to the combined qualification that you want to separate. (To select more than one, hold down the Shift key and click each box.)

2 On the Query menu, click Qualification, then click Uncombine.

Using Negative Qualifications

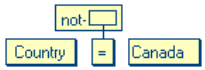
You can make a qualification have the opposite meaning to the condition specified. This is useful when you wish to exclude certain information from a query, or when it is easier to specify the information you don't want than to specify the information you do (for example, if you want all customers located outside Illinois and Indiana).

To negate a qualification:

1 In the qualification tree, select the qualification you want to negate by clicking the corresponding condition box. (To select more than one condition box, hold down the Shift key and click each box.)

To remove the Not indicator, click it.

2 On the Query menu, click Qualification, then click Negate Clause. The Not indicator appears in front of the condition box.



Using Query Modifiers

Query modifiers allow you to change the way a query operates. One use of query modifiers is to summarize information. For example, you may have stores in many different countries. A query that lists these countries can include duplicates if there are several stores in the same country. Alternatively, you may want to know how many stores are located in each country.

There are three query modifiers in BI Query:

- Distinct—Removes duplication and shows you a list of unique values—that is, for the above example, a listing of the unique countries.
- Count All—Lets you find how many times each item occurs in the database. Using the above example, you would get a listing showing the number of times each country name occurs in the database.
- Trim Relationships—Removes from a query any relationship to a data object that has no attributes selected. You can also trim relationships by selecting this feature in the Preferences dialog box.

To use a query modifier:

In an attribute window, before submitting a query, point to Modifiers on the Query menu, then click one of the query modifiers.

Qualifying One Attribute with Another

For more information, see "Creating Calculated Attributes" on page 186.

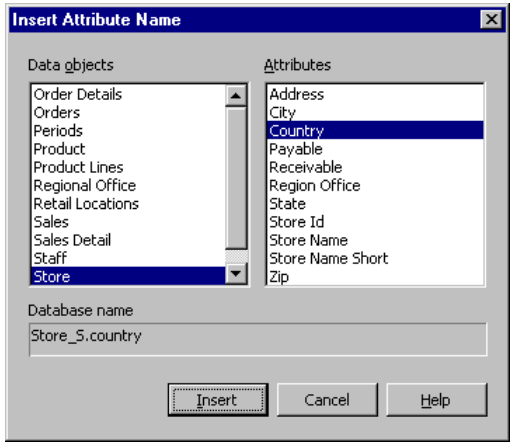
You can build queries in which one attribute qualifies another, so that the limits of the second are applied to the first. The attributes you use as qualifiers can be calculated attributes.

When you use an attribute as a qualifier, you must use the database name of the attribute, not its display name. For example, to get a listing of all profitable products, you define profitability as a unit price at least ten per cent higher than the unit cost (Unit Price > Unit Cost * 1.10). To qualify the Unit Price attribute to find all the products whose unit price is greater than unit cost * 1.10, you cannot type `unit cost * 1.10` in the Qualification box: BI Query would interpret `unit cost` as text, not as an attribute name. You must insert the real database name for attribute Unit Cost.

To qualify one attribute with another:

1 In the qualification tree, click in the Qualification box for the first attribute.

2 On the Edit menu, click Insert Attribute Name. The Insert Attribute Name dialog box opens.



3 From the Data Objects list, select the data object that contains the attribute you want to qualify first.

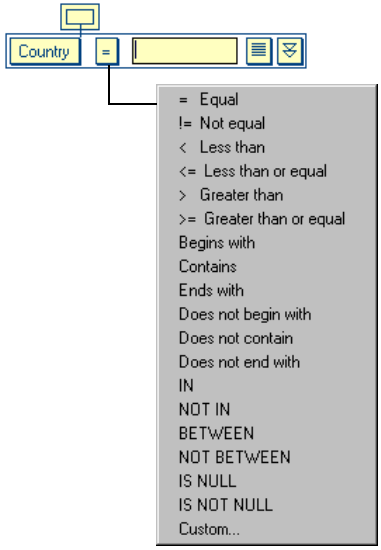
4 From the Attributes list, select an attribute, then click Insert.

5 From the Operator list for the qualified attribute, select the appropriate operator.

6 If you want to include a calculation, click in the Qualification box, and type the calculation. BI Query automatically delimits the expression with brace brackets (for example, `{Sales.order * 100}`).

Using Qualification Operators

When you qualify an attribute, BI Query displays the = operator in the qualification by default, then finds all the records in the database that match the qualification(s) you specify. For example, by setting the Country attribute equal to Spain, the results include only data from that country.



Using the Operator list in the qualification tree, you can apply other operators to restrict a qualification. Keep in mind the following guidelines when applying an operator:

- the >, >=, <, <= operators let you insert a single value when you submit the query

- the BETWEEN and NOT BETWEEN operators require two values (for an upper and lower range)
- the IN, NOT IN, and <> operators let you insert one or more values

Matching Character Patterns

You may not be sure how an item is stored in the database, or you may want to find records for which there is some kind of pattern. For example, you may need information about a retailer and aren't sure of the spelling of the name but know that it begins with "Cars". You may also need a list of businesses with a postal code of 30115.

Examples

Qualification	Possible Matches
Retailer name *Begins with* Sweet	*Sweet* Indulgence, *Sweet*s to Go.
Retailer name *Does not contain* Limited	All retailer names except those containing *Limited*
Zip/Postal code *Ends with* 0210	90210, D-0210, DK-0210

Using Wildcard Characters

You can use wildcard characters with operators such as Begins With, Ends With, or Contains to make less specific searches of the DBMS.

There are two wildcard characters: % represents a character string of any length (including no length); _ represents a single character

Examples

Qualification	Possible Matches
Begins with _r	anything that has r as the second letter
Ends with Smith%	Ms Smith, Smithers, Smithsonian
Contains L%t%d%	Limited, Ltd., Ltd, Laminated

Making Comparisons

When you need to compare values (for example, to find values that are larger or smaller, earlier or later in the alphabet, or equal to a value), you can specify the value on which you want to base the comparison in the Qualification box and apply the appropriate comparison operator. While comparisons are typically applied to numbers, they can also be applied to characters and dates. With characters and dates, the less than symbol (<) means earlier in the alphabet or in chronological order; the greater than symbol (>) means later.

Examples

Qualification	Possible Matches
Credit Limit = 2,000	Credit limits exactly equal to 2,000
Credit Limit != 2,000	All credit limits except those equal to 2,000
Credit Limit < 2,000	Credit limits less than 2,000
Credit Limit <= 2,000	Credit limits less than or equal to 2,000
Credit Limit > 2,000	Credit limits greater than 2,000
Credit Limit >= 2,000	Credit limits greater than or equal to 2,000

Matching Items in a List

When you qualify an attribute with a list, BI Query automatically applies the IN operator to the list. The IN operator allows you to find all records that match any one item in the list. You can also find items that don't match those in a list.

Examples

Qualification	Possible Matches
Country IN France, Germany, Spain	France, Germany, Spain
Country NOT IN France, Germany, Spain	All countries except France, Germany, Spain

Finding a Range of Values

You can qualify an attribute with a range of values to obtain results within a particular range. The range is inclusive, retrieving the lower value and the upper value as well as the values in between.

To qualify with a range of values, enter the values as a qualification list and apply the BETWEEN operator. You can also find results that are outside a particular range by applying NOT BETWEEN. If you qualify a character field, be sure you use the same case (upper or lower) in which the value is stored in the database.

Examples

Qualification	Possible Matches
Credit Limit BETWEEN 2,000 and 5,000	Credit limits between and including 2,000 and 5,000
Credit Limit NOT BETWEEN 2,000 and 5,000	Credit limits outside this range

Matching Any/Matching All

You can enter more than one value in the Qualification box and choose to match any of the values or all of them.

To match any or match all values:

1 With a data model open, on the Query menu, click New.

2 Double-click a data object.

3 In the attribute window, select an attribute and select the Qualify box in the same row.

4 In the qualification tree, from the Operator list for the selected attribute, select one of the following operators:

- Begins With
- Does Not Begin With
- Contains
- Does Not Contain

- Ends With
- Does Not End With

5 In the Qualification box, type the values that you want to match. Press the down arrow after entering each value.

6 In the List box, select either Match Any or Match All.

7 On the Query menu, click Run. Alternatively, on the Application toolbar, click the Run Query button.

Determining if Data is Missing

A NULL value occurs when no value has been entered into the database. It doesn't represent a zero or a character space but is instead a placeholder for missing information. Because NULLs don't match anything, you can't retrieve them using standard comparison qualifications such as Greater Than or Less Than. To include or exclude rows for which information hasn't been recorded, apply the IS NULL or IS NOT NULL operator to an attribute.

To include or exclude rows for which information hasn't been recorded:

1 In the attribute window, select an attribute and select the Qualify box in the same row.

2 In the qualification tree, from the Operator list for the selected attribute, select the IS NULL operator to include rows with NULL values or the IS NOT NULL operator to exclude rows with NULL values.

3 On the Query menu, click Run. Alternatively, on the Application toolbar, click the Run Query button.

In some cases, missing information is represented by a character space rather than a NULL (if, for example, the space bar was pressed during data entry). Since a space is a character to BI Query, the above method will not work. In this case, you have to query for {' '}.

To find rows containing a character space:

Combining the two queries with an OR will allow you to test if information is missing regardless of whether it is a NULL condition or a space.

1 In the attribute window, select an attribute and select the Qualify box in the same row.

2 In the qualification tree, from the Operator list for the selected attribute, select the = operator.

3 Type {' '} in the Qualification box. Be sure to put a space between the single quotation marks.

4 On the Query menu, click Run. Alternatively, on the Application toolbar, click the Run Query button.

Applying Your Own Operators

In addition to applying the standard SQL operators provided in BI Query, you can restrict a qualification using any other operator that your DBMS (database management system) supports. (For information about operators your DBMS supports, see the documentation for your DBMS.)

To apply your own operator:

1 In an attribute window, select Custom from the Operator list in the qualification tree.

2 In the Custom Operator dialog box, type an operator into the Operator box.

3 Click OK.

Querying Multiple Data Objects

You can create a query involving more than one data object, as long as a relationship exists between the data objects. BI Query depicts relationships as lines drawn between data objects in a design window.

If there is no relationship between two data objects, you can create a dynamic relationship between them. For more information, see "Creating Dynamic Relationships" on page 197.

To query multiple data objects:

1 Double-click each object in turn and select the attributes that you want.

The more objects that are used in a query, the longer the query will take.

If the objects are directly joined, BI Query automatically highlights the relationship between them. If there are any other objects between the two you want to access, select the attributes from the first object, click the lines linking it to the second data object, then select the attributes from the second object. This allows BI Query to make the necessary adjustments to your query.

As you build your query, the qualification tree at the bottom of the attribute window displays all qualifications you have added to the query, regardless of the attribute on which they were placed.

2 When you have finished building the query, click the Run Query button on the Application toolbar.

Removing Unnecessary Joins from Queries

For more information on join conditions, see the *BI Query Data Models User's Guide*.

The relationship between two data objects consists of one or more join conditions that specify how the data objects are related. When you submit a query that involves both objects, any rows returned from the DBMS must satisfy the join conditions between the objects.

You can set the Trim Relationships preference to remove excess joins from the SQL for a query. If a query contains a relationship to a data object that has no attributes selected, the Trim Relationships preference removes the relationship when you submit the query. If you want to add qualifications by selecting a relationship, you must disable this preference.

To set the Trim Relationships preference, do one of the following:

- On the Tools menu, click Preferences. In the Preferences dialog box, select Trim Relationships.
- Open or create a particular query. On the Query menu, point to Modifiers, then click Trim Relationships (if it is not already selected).

Updating Queries

After the administrator has updated a data model, some of the saved queries may not work—for example, they may include attributes that no longer exist in the database. When you open a data model, BI Query compares the dates on which the data model and the queries were last refreshed. If the data model was refreshed more recently than the queries, BI Query prompts you to refresh them.

The administrator may have also changed the syntax type for outer joins in the data model, in which case the saved queries that include those joins may no longer work correctly. If you enable the Preserve Legacy Query Outer Joins preference, BI Query treats new-syntax outer joins as legacy joins when you run a saved query.

Refreshing a Query

When you refresh a query, BI Query compares the queries with the data model and displays information on each query that has been affected by changes to the data model. BI Query can detect only whether attributes have been deleted, and not whether they have been changed.

To refresh queries:

1 Ensure that BI Query is not in Design Mode.

2 On the Tools menu, click Refresh Queries.

3 In the Refresh Query dialog box, select the queries you want to refresh (or click Select All), then click Refresh. (Data values queries are identified by the words Data Values after the query name. Data values queries cannot be refreshed.)

4 Do one of the following:

- Click Continue to leave the query unchanged and view the next one.
- Click Change to update the query.
- Click Change All to update all the queries selected.
- Click Cancel to return to the Refresh Query dialog box without comparing the remaining queries.

If you modify any queries, BI Query makes the changes, then saves the queries in the `Queries` folder. BI Query also saves the original queries with the extension `.bak`.

5 Submit any modified queries to make sure they still produce the desired results.

Using Legacy Joins As Is

When you run a query that uses legacy outer-join operators in a data model that has been converted to a new outer-join syntax, BI Query automatically converts the query to the new syntax. If this conversion does not produce the same results as the query using the old operators, you can force BI Query to treat all outer joins in the data model as if they used the Classic (ANSI SQL/89) syntax.

To use Classic syntax for legacy joins in a query:

1 On the Tools menu, click Preferences.

2 In the Preferences dialog box, select Preserve Legacy Query Outer Joins.

3 Click OK.

> **Note:** The Preserve Legacy Query Outer Joins preference does not override the Preserve Legacy Query Outer Joins permission set by the administrator.

Chapter 4

Formatting Data

About Data Formats **57**
Data Type Categories 57
Data Format Options 58
Format Order 59

Formatting Data: The Basics **60**
Opening the Format Dialog Box 61
Specifying Data Model Default Formats 63
Setting the Null Values Option 64
Setting the String Option 65
Setting the Computed String Option 66

Formatting Specific Data Types **68**
Formatting Boolean Data 68
Formatting Numeric Data 69
Formatting Date and Time Data 71

Formatting Data for Data Entry **72**
Setting the Data Entry Format 73
Entering Date and Numeric Data 73
Converting Year Formats 74
Correcting Invalid Entries 75

About Data Formats

A data format specifies how BI Query represents data in a result set or report. You can apply a data format to any of the following:

- Attributes—Whenever you run a query that involves the formatted attribute, BI Query applies the format to the corresponding column in the results set (and report, if you create a report based on the results set).
- Columns in a results set or report—In this case, you format the data *after* you have run a query. The format you specify is limited to the current result set or report and does not apply to any other.
- Data types in the data model—In this case, the format you specify for a particular data type applies to all attributes (and therefore results) of that data type in the model. This format is the default data model format for the data type.

> **Note:** The format applied to a particular attribute or result/report column overrides the data model format for the corresponding data type.

Data Type Categories

For formatting purposes, BI Query organizes the various data types in the data model into general categories. For example, BI Query treats the distinct integer data types (Tinyint, Smallint, Integer, Largeint, and so on) as one formatting category.

The available categories are as follows:

- Boolean, for boolean or bit data
- Currency, for numeric data that represents currency values
- Date, for date-and-time, date-only, and timestamp data
- Fixed, for numeric data with a fixed scale (a fixed number of digits after the decimal point)
- Float, for floating-point (real) data
- Integer, for integer data of any standard byte size

- Null, for NULL data
- Time, for time-only data

> **Note:** The Null category does not represent a particular data type but rather the NULL values of *any* data type.

In Windows 2000, default data formats are defined in the Regional Options item in the Control Panel.

With the exception of the Boolean category, BI Query takes the initial default format for each category from the Regional Settings item in the Windows Control Panel. (There is no Regional Settings value for boolean data.)

Use the Format dialog box to set the format for a category. The appearance and functionality of the dialog box depends on the category you want to format.

Data Format Options

For each data type category, BI Query provides the following formatting options:

Format Option	Description
Category-specific options	The formatting options that are specific to the category. For example, for numeric data types, you can specify the scale—the number of digits after the decimal point. BI Query also provides a set of predefined formats for each category that you can use to format a particular attribute, column, or data model default. The predefined formats include the Regional Settings defined in the Control Panel. If you are formatting an attribute or result/report column, you can also select the default data model format.
String	A template for the category in the form of a string. Using special control characters in the string, you can add or remove characters in all values of the selected attribute or result column. For example, you could format a string of 10 digits into a recognizable telephone number. This option is not available for data model formats.

Format Option	Description
Null values	The format for any NULL values in the category. This format is specific to NULL values in the category; it is not the same as the default format for the Null data type category.
Computed string	A conditional replacement string for values in the category. You can change the formatting color and value of data in the selected attribute or result column based on a comparison between the original data and one or more specified values. For example, you could replace all values less than a certain number with the word "Unsatisfactory". This option is not available for default data model formats.

Format Order

You can set more than one formatting option for a given attribute, column, or data model default. However, the particular options that apply depend on the value of the data.

Format Order for NULL Data

If the data is NULL, BI Query applies the Null Values formatting only; no other formatting options apply. The format that BI Query uses in this case is one of the following (in order of precedence):

1. the Null Values format for the attribute or result column (if specified)
2. the data model format for Null Values in the corresponding data type (if specified)
3. the data model format for the Null data type category

Format Order for Non-NULL Data

If the data is non-NULL, BI Query applies the specified options in the following order:

1 If you have used the Computed String option, BI Query evaluates the conditions based on the value of the data. If the data meets the conditions, BI Query applies the Computed String formatting; no other options apply in this case.

2 If you have not used the Computed String option, or if the data does not match the conditions defined for it, BI Query applies one of the following category-specific formats (in order of precedence):

   a) the formatting for the attribute or result/report column (if specified)

   b) the default format for the corresponding data type

3 If you have specified String formatting, BI Query applies it to the result from step 2.

Formatting Data: The Basics

The following procedure outlines the basic steps to follow in setting a data format for one of the following targets:

- an attribute
- a result/report column
- a data model default

You can find more detailed information on each step later in this chapter.

To specify a data format for a particular target:

1 Open the Format dialog box. The topmost item in the Format Type list displays the data type category for the selected target. For more information, see "Opening the Format Dialog Box" below.

2 Do one of the following:

- To use a custom format, specify the category-specific options for the target. For more information, see "Formatting Specific Data Types" on page 68.
- To use the Regional Settings for the category, select <Regional Settings> from the Predefined Formats list.
- To use the default data model format for the category, select Default (available only for attributes and result/report columns).

3 To format NULL values for the target, select Null Values from the Format Type list, then specify the format. For more information, see "Setting the Null Values Option" on page 64.

4 To format the target as a string, select String from the Format Type list, then specify the format. For more information, see "Setting the String Option" on page 65.

5 To format the target according to certain conditions, select Computed String from the Format Type list, then specify the format. For more information, see "Setting the Computed String Option" on page 66.

6 Click OK. BI Query formats the data for the target according to the options you have specified.

Note: You can use more than one format option, but BI Query applies them according to its format order rules. For more information, see "Format Order" on page 59.

Opening the Format Dialog Box

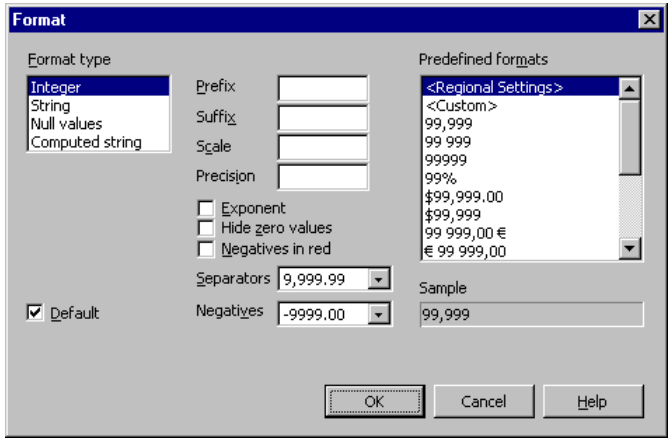
The procedure for opening the Format dialog box depends on the target you want to format.

To format an attribute:

1 Open the attribute window for the data object that stores the attribute. (BI Query must not be in Design Mode).

2 Click the question mark button beside the attribute name. The Attribute dialog box opens.

3 Click Edit. The Format dialog box opens.



The Format dialog box for Integer data types

To format a column in a results set:

1 Click the column you want to format.

2 On the Results menu, click Format. The Format dialog box opens.

To format a column in a standard report:

1 In the standard report, click the column you want to format.

2 On the Report menu, click Format. The Format dialog box opens.

To format a data model default:

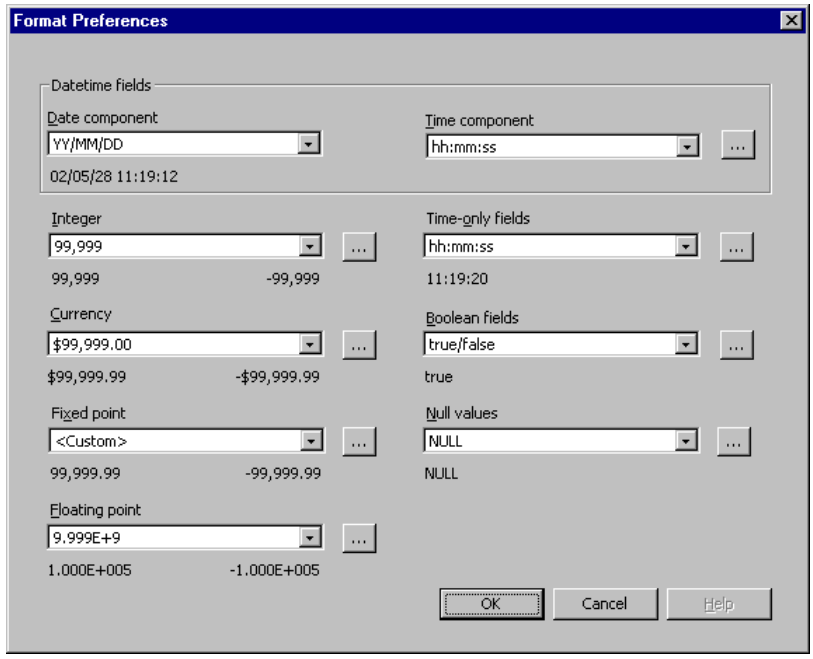
1 On the Tools menu, click Preferences.

2 In the Preferences dialog box, click Data Formatting. The Format Preferences dialog box opens.

3 Click the ellipsis (...) button for the data type category you want to format. The Format dialog box opens.

Specifying Data Model Default Formats

Use the Format Preferences dialog box to specify the default formats for the entire data model. These defaults apply to any target (attribute or result/report column) that does not have its own format.



Note: A data model can connect to more than one DBMS. The data type for a given attribute is determined by the DBMS and the connection to it. However, the default formats you specify for the data model are not connection specific; they apply to all design windows and, therefore, all connections.

To specify the data model format for a data type category:

1 On the Tools menu, click Preferences.

2 In the Preferences dialog box, click Data Formatting.

3 In the Format Preferences dialog box, do one of the following:

- To use a predefined format, select it from the drop-down list that corresponds to the category. To use the Regional Settings defined in the Control Panel, select <Regional Settings>.
- To specify a custom format, select <Custom> from the drop-down list, then click the ellipsis (...) button. In the Format dialog box, specify the custom setting, then click OK. For more information, see "Formatting Data: The Basics" on page 60.

4 Repeat step 3 for any other category you want to format.

5 Click OK.

Setting the Null Values Option

You can format NULL values for a particular attribute, result/report column, or data type. You can also set the default format for all NULL values in the data model.

To specify NULL formatting for an attribute, column, or data type:

1 Open the Format dialog box for the target you want to format. For more information, see "Opening the Format Dialog Box" on page 61.

2 From the Format Type list, select Null Values.

3 Do one of the following:

- To use a predefined format, select it from the Predefined Formats list.
- To specify your own display format for NULL values, type it in the Display box.

4 Click OK.

To specify the data model format for NULL values:

1 On the Tools menu, click Preferences.

2 In the Preferences dialog box, click Data Formatting.

3 In the Format Preferences dialog box, do one of the following:

- To use a predefined format, select it from the Null Values drop-down list. To use the Regional Settings defined in the Control Panel, select <Regional Settings>.
-
To specify a custom format, select <Custom> from the drop-down list, then click the ellipsis (...) button. In the Format dialog box, specify the custom setting, then click OK.

4 Click OK.

Setting the String Option

Use the String option to format any data type as a string. This option lets you include additional information in the data (such as a punctuation mark or a code) and exclude unwanted information.

To use the String option, you create a template string for the data; the template can contain special control characters that dictate the data format. The following special characters are available:

Character	Description
#	Inserts a character from the original data (in sequence).
_	Suppresses a character in the original data (in sequence).
*	Inserts the remaining characters in the original data.
\	Inserts the next character as is. Use this character to include one of the special formatting characters as itself. For example, type \# to insert the # symbol.

To insert any other character, type it where you want it to appear in the string.

To specify the String option:

1 Open the Format dialog box. The String option is available only for formatting attributes and result/report columns. For more information, see “Opening the Format Dialog Box” on page 61.

2 From the Format Type list, select String.

3 In the Format String box, type the template string you want to format the data. Use the special characters described above.

4 Click OK.

5 If you are specifying the option for an attribute, click OK in the Attribute dialog box.

Example

A database attribute stores telephone numbers as 13-character strings (for example, 6135554355x26). To format the data in this attribute as a recognizable phone number, you can create the following template:

```
(###) ###-####_ Ext. \#*
```

If you apply the String option using this template, BI Query formats the value

```
6135554355x26
```

as

```
(613) 555-4355 Ext. #26
```

Setting the Computed String Option

The Computed String option lets you replace values in the target (an attribute or result/report column) based on certain conditions. For example, to identify areas that require further attention, you could replace certain values in the target (all sales below a certain amount) with the word "Unsatisfactory". If the target is a standard report, you could also apply a color to these values.

Each condition for the Computed String option has the following components:

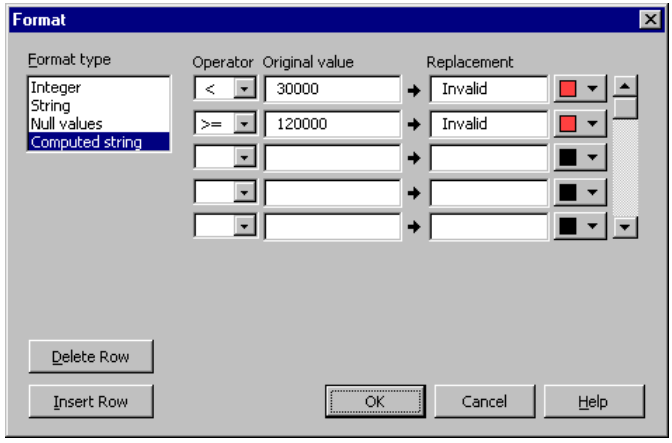
- A test value—BI Query compares each value in the target against this value.
- An operator—The operator specifies the nature of the comparison. For example, if you select < as the operator, BI Query checks if each value in the target is less than the test value.

- A replacement value—If the result of the comparison is true, BI Query substitutes the replacement value for the value in the target.
- A replacement color (optional)—If the target value meets the condition and the results appear in a standard report, BI Query applies the specified color to the replacement value.

BI Query evaluates and applies each condition separately.

To specify the Computed String option:

1 Open the Format dialog box. The Computed String option is available only for formatting attributes and result/report columns. For more information, see "Opening the Format Dialog Box" on page 61.



2 In the Format dialog box, select Computed String from the Format Type list.

3 Place the cursor in the first empty row of conditions.

4 From the drop-down list in the Operator column, select the operator you want to use to compare the results value against the test value.

> **Note:** You can apply the Else operator to the last condition only.

5 In the box in the Original Value column, type the test value.

6 To specify a replacement value, type it into the box in the Replacement column.

7 To specify a color for the new value (in standard reports only), click the color button in the same row and specify a color. If you have not specified a replacement value, BI Query applies the color to the value in the target.

8 To specify more than one substitution condition, repeat steps 3–7 for each condition.

9 Click OK.

Note: To insert a row, place the cursor in the row below which you want to insert the new one, then click Insert Row. To delete a row you've specified, place the cursor in the row, then click Delete Row.

Formatting Specific Data Types

For more information on setting the default formats, see "Specifying Data Model Default Formats" on page 63.

Attributes and result/report columns have specific data types. However, for formatting purposes, BI Query collects related data types into general data categories. When you format an attribute or a result/report column, the available formatting options depend on the category for the corresponding data type. Similarly, you specify the default formats for the data model in terms of categories and not separate data types.

The following sections describe the formatting options available for each category.

Formatting Boolean Data

The boolean or bit data category stores one of two values. The names applied to the two values depend on the context: yes or no, true or false, 1 or 0, and so on. You can set these names as part of the format.

To format boolean data:

1 Open the Format dialog box for the target you want to format. For more information, see "Opening the Format Dialog Box" on page 61.

2 Do one of the following:

- From the Predefined Formats list, select the format you want to apply.
- In the True and False boxes, type the formats for the corresponding boolean values.

3 Apply any other options you want. For more information, see "Formatting Data: The Basics" on page 60.

4 Click OK.

5 In the Attribute dialog box or Format Preferences dialog box, click OK (if necessary).

Formatting Numeric Data

The numeric data categories (Currency, Fixed, Float, and Integer) use the same version of the Format dialog box. The following table describes the category-specific options for numeric data:

Option	Description
Prefix	Specifies the character or set of characters that precedes the numeric value (such as a dollar symbol, $).
Suffix	Specifies the character or set of characters that are appended to the numeric value (such as a percent sign, %).
Scale	Specifies the scale for the numeric value (the number of digits after the decimal point).
Precision	Specifies the total number of digits in the number (including the digits after the decimal point but excluding any prefix and suffix characters). If the actual precision of a number is smaller than the specified precision, BI Query pads the number with leading zeros. If the actual precision is larger than the specified precision, BI Query preserves the original precision of the number.

Option	Description
Exponent	Permits numbers with exponents. If you set this option, you cannot edit qualification values.
Hide Zero Values	Removes zero values from the results.
Negatives In Red	Displays negative values in red (in standard reports only).
Separators	Specifies the position and format for separators (such as commas and periods).
Negatives	Specifies the text format for negative numbers.

Note: Not all options are meaningful for each category.

When you set the format for numeric data, you can either use a predefined format or customize your own.

To use a predefined format for numeric data:

1 Open the Format dialog box for the target you want to format. For more information, see "Opening the Format Dialog Box" on page 61.

2 Do one of the following:

- From the Predefined Formats list, select the format you want to use. To use the Regional Settings for the category, select <Regional Settings>.
- To use the data model format for the category, select Default (available only if you are formatting an attribute or result/report column).

3 Apply any other options you want. For more information, see "Formatting Data: The Basics" on page 60.

4 Click OK.

5 In the Attribute dialog box or Format Preferences dialog box, click OK (if necessary).

To specify a customized format for numeric data:

1 Open the Format dialog box for the target you want to format.

2 To modify an existing format, select it from the Predefined Formats list.

3 Specify any of the options described above.

4 Apply any other formatting options.

5 Click OK.

6 In the Attribute dialog box or Format Preferences dialog box, click OK (if necessary).

Formatting Date and Time Data

The Date data category encompasses date-and-time, date-only, and timestamp data. The Time category is for time-only data. You can format both categories using the same version of the Format dialog box.

To format date or time data:

1 Open the Format dialog box for the target you want to format. For more information, see "Opening the Format Dialog Box" on page 61.

2 Do one of the following:

- To use a predefined time format, select it from the Predefined Time Formats list.
- To remove the time component from date-and-time data, select <Blank> from the Predefined Time Formats list.
- To use the Regional Settings for the category, select <Regional Settings> from the Predefined Time Formats list.
- To use the data model format for the category, select Default (available only if you are formatting an attribute or result/report column).

3 If you are formatting Date-category data, do one of the following:

- To use a predefined date format, select it from the Predefined Date Formats list.

- To remove the date component from date-and-time data, select <Blank> from the Predefined Time Formats list.
- To use the Regional Settings for the category, select <Regional Settings> from the Predefined Time Formats list.
- To use the data model format for the category, select Default (available only if you are formatting an attribute or result/report column).

4 To modify the selected format, edit the text in the Format String box.

5 Apply any other options you want. For more information, see "Formatting Data: The Basics" on page 60.

6 Click OK.

7 In the Attribute dialog box or Format Preferences dialog box, click OK (if necessary).

Formatting Data for Data Entry

The data format you specify for a given attribute governs how BI Query formats values for that attribute in a results-set column or report column. This format also applies when you enter data in a qualification or prompt for that attribute, or when you enter data to update the database values for that attribute.

BI Query can recognize multiple formats for a given data type. When you enter data, if BI Query is able to interpret the format you use, it automatically converts the data to the specified format for the attribute. If you enter data in a format that BI Query is unable to interpret, it displays an alert message.

> **Warning!** Avoid using an attribute format for numbers or dates that "hides" information. For example, if a date attribute has the format MMM/DD and you type `Sept 15, 2001` as the qualification value, BI Query inserts '2001-09-25' into the SQL for the query, but displays Sep/15 in the qualification tree. If you change the value to `Sept 26`, BI Query interprets your change as being a complete value (Sept 26, not Sept 26, 2001). The "hidden" year is lost, and the date value in the SQL ('-09-26') becomes invalid.

Setting the Data Entry Format

By default, BI Query takes the data entry format for an attribute from the data format for that attribute. However, you can set the data entry format to the data model default for the corresponding data type.

To set the data entry format:

1 On the Tools menu, click Preferences. The Preferences dialog box opens.

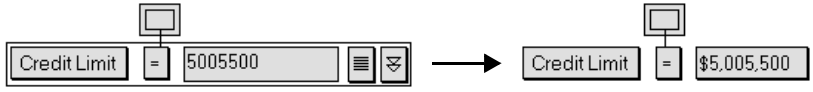
2 In the Qualifications area, do one of the following:

- To use attribute formats for data entry, select Attribute.
- To use the data model formats for data entry, select System Default.

3 Click OK.

Entering Date and Numeric Data

When entering data, you can type numbers (including currencies) without formatting them. BI Query converts them to the format you've specified for the corresponding attribute.



You can also type dates using any format comparable to the one you've specified. For example, if the date format you've specified is MM/DD/YYYY, you can type dates in the following formats:

- `2001.1.14`
- `1.14.2001`
- `January 14, 2001`
- `2001 Jan 14`

As long as you include the day, month, and year and BI Query is able to unambiguously interpret the format, it converts the date to the specified format.

Note: BI Query cannot reliably interpret ambiguous dates such as 04/05/2002, in which the day can be confused for the month, or 04/05/02, in which the day, month, and year are not obvious. To ensure that BI Query interprets dates correctly, make sure that you type dates using all four digits of the year and, if necessary, use the name of a month rather than its number.

Converting Year Formats

You can specify a date window to control how BI Query converts two-digit years to four-digit years.

The date window consists of a start year and end year. If you specify a date window with a start year of 1918 and an end year of 2017, BI Query adds the century 19 to all two-digit years greater than or equal to 18 and the century 20 to all two-digit years less than or equal to 17. When you enter 45, BI Query converts it to 1945; when you enter 15, BI Query converts it to 2015.

Note:

- If you change the date window after you've entered two-digit years, BI Query doesn't apply the new date window to those dates.
- Changes you make to the date window in one Hummingbird BI product affect all other Hummingbird BI products.

To specify the date window:

1 On the Tools menu, click Date Entry. The Date Entry dialog box opens.

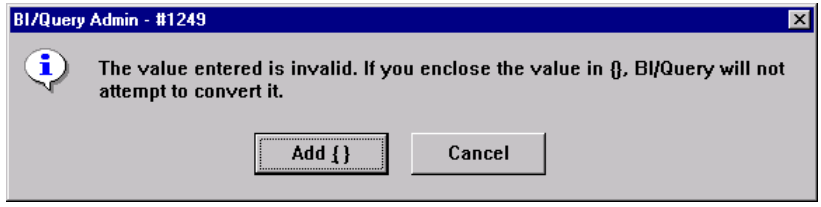
2 In the Start Year Is box, type the start year for the date window. (The end year is calculated automatically.)

3 Select Sliding or Fixed. A sliding date window adds one to the start and end year on January 1st every year. For example, if the date window is 1918 to 2017, it becomes 1919 to 2018 when the current year changes. A fixed date window is not incremented automatically.

4 Click OK.

Correcting Invalid Entries

BI Query alerts you if you type data in a format that it can't interpret.


BI/Query Admin - #1249
The value entered is invalid. If you enclose the value in {}, BI/Query will not attempt to convert it.
Add { }
Cancel

To return to an entry and correct it:

Click Cancel in the alert window.

To force BI Query to use the entry you typed:

This feature is for users with advanced knowledge of SQL database access.

Click Add { }. Your entry appears within curly brackets in the qualification tree. When you submit a query, BI Query inserts your data in the SQL string as is.

To determine the format for an attribute:

For more information on configuring data entry formats, see "Setting the Data Entry Format" on page 73.

1 If you have configured the data model to use the default data model formats for data entry, open the Format Preferences dialog box:

   a) On the Tools menu, click Preferences.

   b) In the Preferences dialog box, click Data Formatting.

   Note the current default format for the relevant data category.

2 If you have configured the data model to use attribute formats for data entry, open the Attribute dialog box for the relevant attribute. (In the attribute window, click the question mark button beside the name of the attribute).

   The Format box displays sample data in the attribute format.

Chapter 5

Using Data Values Files

About Data Values Files **79**
Types of Data Values Files 80
Naming and Storing Data Values Files 80
Order of Data Values File Use 81

Creating Data Values Results Files **82**
Creating Data Values Results Files Using a Text Editor 83

Creating Data Values Query Files **85**

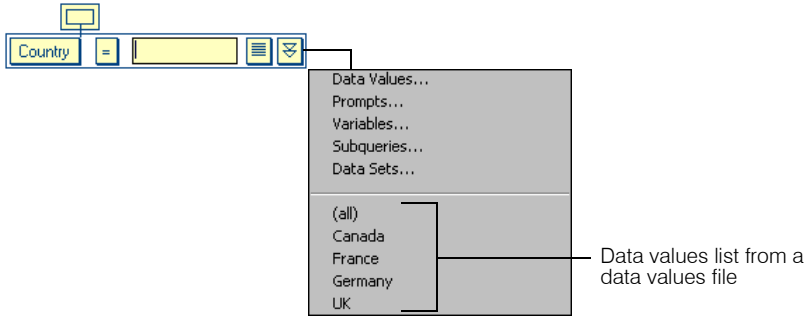
Using a Data Values File in a Query **86**

Data Aliasing **87**
Creating a Data Values File to Alias Data 88

About Data Values Files

One way to limit what a query retrieves is by qualifying attributes with data values. Before you submit the query, you can insert one or more data values into a qualification, either by typing them or by choosing them from a list. Lists of data values are provided by data values files.

A data values file contains a subset of the possible values in the database. These values appear as a list when you click the Data Values button. Selecting data values from a list is a more accurate and efficient way to insert values than typing them.



If no data values file is available for a particular attribute, when you click the Data Values button and select Data Values, BI Query runs the default data values query and retrieves all the values.

The administrator can disable access to the default data values query. Disabling it is recommended when loading all the values in the database uses network resources inefficiently—for example, when an attribute has a large number of values or when those values are simply numbers and dates.

Data values files can also be inserted into conditional prompts for use in executive information system buttons. They can also be used to set up data aliasing, making it possible, for instance, for users to select items by meaningful names instead of by numerical codes.

Types of Data Values Files

There are two types of data values files:

Data Values Query Files Query the database and retrieve a subset of values to create the list. Data values query files provide the most accurate lists; the first time you use them in a session, they query the database to supply the values. Because they can return different results depending on when they are run, they are sometimes called dynamic data values files.

Data Values Results Files Display a stored set of values. Data values results files are produced by storing the results of a query or, sometimes, are produced manually. The lists of values they contain are not updated by BI Query, so they are sometimes called static data values files. Results files reduce the load on the database server because they eliminate the need to query the database.

For more information, see "Publishing Data Values Results Files" on page 243.

If your querying environment includes BI Server, you can let other users access your data values results files by publishing them to the BI Server Repository.

> **Note:** Data values results files are useful only when data values change infrequently or not at all (for example, sales data from previous years). If they must be updated often, it is more efficient to create data values query files instead.

Naming and Storing Data Values Files

For more information, see "Naming Conventions for Prompts" on page 94.

Data values query and results files require special naming conventions. When they supply values to be used in a prompt, they use the prompt name. When they're used in both qualifications and prompts for a particular attribute, they use the following format:

```
data_object.attribute
```

where

- *attribute* is the name of the attribute
- *data_object* is the name of the data object that stores the attribute.

data_object and *attribute* are the names used in the data model (not the names of the table and column in the database). For instance, for a data object called Retailers with an attribute called country, a data values file would be called `Retailers.country`. Be sure to use the case used in the attribute or prompt name.

Data values query and results files are stored in the `DataVals` folder. BI Query stores data values results files in two separate files: the `.qrd` file, which stores results in text format, and the `.qrr` file, which stores additional information that the application needs to read the text file. Without the `.qrr` file, BI Query can't open the `.qrd` file. Data values query files have the same file extension as regular queries, `.qry`.

For information on the order in which data value files are used, see "Order of Data Values File Use" on page 81.

The file names you give these files determine the order in which BI Query selects the files in the `DataVals` folder as well as whether the corresponding data values are available in a qualification or in a prompt.

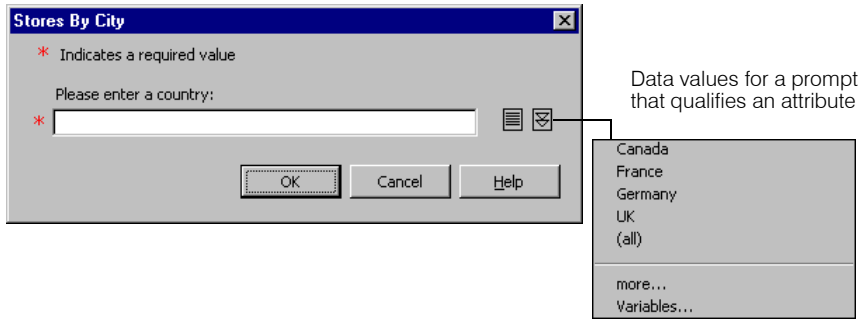
Order of Data Values File Use

The following table shows the order in which BI Query uses data values files, their names, and where they're used.

Order	Data Values File	Available In
1	Results file named after the corresponding prompt	Prompt dialog boxes
2	Query file named after the corresponding prompt	Prompt dialog boxes
3	Results file named after the data object and attribute used in the query	Qualification tree and prompt dialog boxes
4	Query file named after the data object and attribute used in the query	Qualification tree and prompt dialog boxes

For more information on prompts, see "About Prompts" on page 91.

When you click the Data Values button in a prompt dialog box in order to display a list of data values that can be used in the prompt, BI Query checks to see if there's a data values file with the same name as the prompt. If there is, BI Query uses it; otherwise, it checks to see if there is a data values file named after the data object and attribute used in the query. If there is, BI Query uses it.



If BI Query doesn't find a data values file, and if data values for the qualified attribute haven't been disabled, you can click Data Values in the pop-up menu to open the Data Values dialog box. The dialog box lists all the unique values in the database for the attribute.

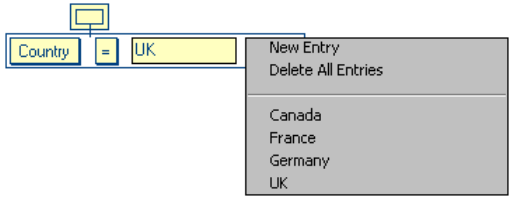
Creating Data Values Results Files

For information on retrieving and opening stored data values results files, see the Help for your BI Query application.

You can create data values results files from BI Query results, or customize the results using a text editor or spreadsheet. You can qualify attributes by inserting these data values into qualifications and prompts.

To create a data values results file:

1. In an attribute window, click an attribute, then click the Sort box and apply any other restrictions you want.
2. On the Query menu, point to Modifiers, then click Distinct to eliminate duplicate values.
3. If you want to retrieve only specific values, qualify the attribute accordingly.



4 Submit the query.

5 If you plan to customize your file using a text editor or spreadsheet, on the Results menu, click Options. Select <CR><LF> from the Record drop-down list box and <None> from the End of File drop-down list box, then click OK.

6 On the Results menu, point to Save As, then click Data Values. The Save Data Values Results dialog box opens.

7 In the Data Values Results Name box, type the file name you want. BI Query adds the extension .qrd to the file. To make the data values in the file available in qualifications and prompts for a particular attribute, name the file using the following format:

object.*attribute*

where

- *object* is the display name of the data object that stores the attribute
- *attribute* is the display name of the attribute

To make the results available only for a specific prompt, use the name of the prompt. Prompt, attribute, and object names are case sensitive.

8 Click Save.

Creating Data Values Results Files Using a Text Editor

You can use any text editor or spreadsheet program to create a data values results file that contains a set of customized data values. To do so, you first create a data values results file to serve as a template for the customized data values. (This sets up the .qrd file, which stores results in text format, and the .qrr file, which stores additional information that the application needs to read the text file.) Once you create these files, you replace the data values in the .qrd file with the customized data values.

When you type values into the text editor or spreadsheet program, type them in the order and case you want them to appear in BI Query, pressing Enter after each value.

To create a data values results file using a text editor:

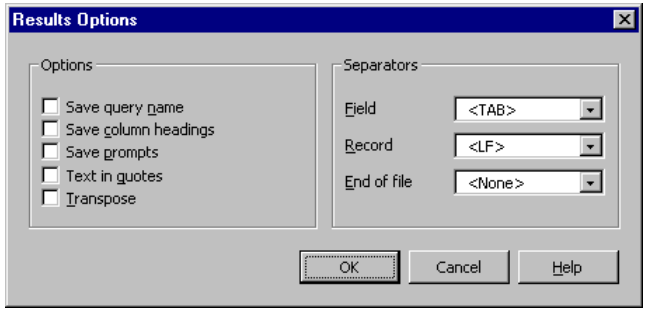
1 In an attribute window, click an attribute. (You can select only one attribute for a data values results file.)

2 Click the Sort box to sort the results, then apply any other restrictions you want.

3 On the Query menu, click Modifiers, then Distinct to eliminate duplicate values.

4 On the Query menu, click Run to submit the query.

5 When the results appear, on the Results menu, click Options. The Results Options dialog box opens.



6 In the Separators area, select <CR> from the Record drop-down list and <None> from the End of File drop-down list.

7 Click OK.

8 On the Results menu, point to Save As, then click Data Values.

9 In the Save Data Values Results dialog box, name the file.

10 Click OK.

11 In a text editor or spreadsheet program, do one of the following:

- Open the .qrd file you created, modify it, and save it under the same name.
- Type a list of values in a new document file, adding a newline (carriage return + line feed) after each one, then save the file using the same path and file name as the .qrd file.

12 To test the file in BI Query, select the attribute used to create the original results, click its Qualify box, then click the Data Values button in the qualification tree.

Creating Data Values Query Files

You can create query files that retrieve subsets of values for the attribute and use them to qualify attributes in other queries by inserting the data values query file into qualifications and prompts.

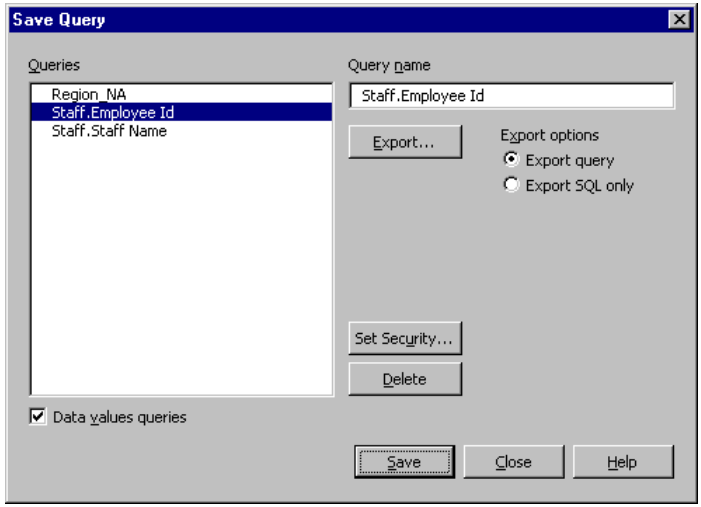
To create a data values query file for an attribute:

1 Double-click the data object that contains the attribute. In the attribute window, select the attribute, then click its Sort box and apply any other restrictions you want.

2 To eliminate duplicates, on the Query menu, point to Modifiers, then click Distinct.

3 Click in the Qualify box beside the attribute, and then enter the values you want in the Qualification box for the selected attribute.

4 On the Query menu, click Save. The Save Query dialog box opens.



5 Select Data Values Queries.

6 In the Query Name box, type the file name you want. A file name with the extension `.qry` is created by BI Query. Use the following file naming convention format to make the data values available in qualifications and prompts:

data_object.attribute

where

- *data_object* is the display name of the data object that stores the attribute
- *attribute* is the display name of the attribute to which you want to apply the file

Case is important—make sure that you match the case of the data object as it appears in the title bar of the attribute window, and the case of the attribute as it appears in the attribute list in the attribute window.

7 Click Save.

Using a Data Values File in a Query

You can use a data values file to qualify an attribute in a query.

To use a data values file in a query:

1 Create a query that involves the attribute the data values file is based on.

2 To access the data values list, click the attribute's Qualify box. Then click the Data Values button in the qualification tree and select Data Values from the pop-up menu.

If you created a data values results file, BI Query loads it and displays the values as a list. If you created a data values query file, BI Query submits it, then displays the values in the Data Values dialog box.

You can also access the data values list by clicking the Data Values button in the prompt window and selecting Data Values from the pop-up menu.

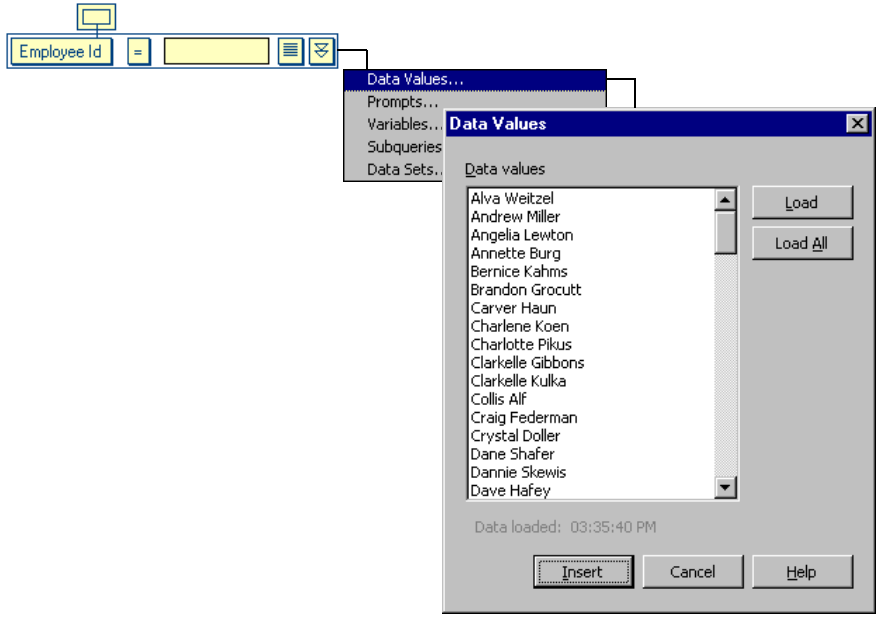
Data Aliasing

Data aliasing lets users select from a list of common-language names for data values while BI Query inserts the corresponding actual values into the query.

To set up data aliasing, you create a data values results file or data values query file that contains two columns; in the first column are the actual values, in the second are the aliases. Only the data in the second column is displayed in the list of data values.

For example, instead of selecting a value for a qualification that may be difficult to understand or remember, such as a staff person's ID number, you'll be able to select a more meaningful value, such as the person's full name. You select the name from a list of data values and BI Query inserts the ID number into the Qualification box for the query.

When you qualify the Employee ID attribute, click the Data Values button in the qualification tree and BI Query displays staff names as a list. When you select a name from the list, BI Query inserts the corresponding ID number into the qualification tree.



Creating a Data Values File to Alias Data

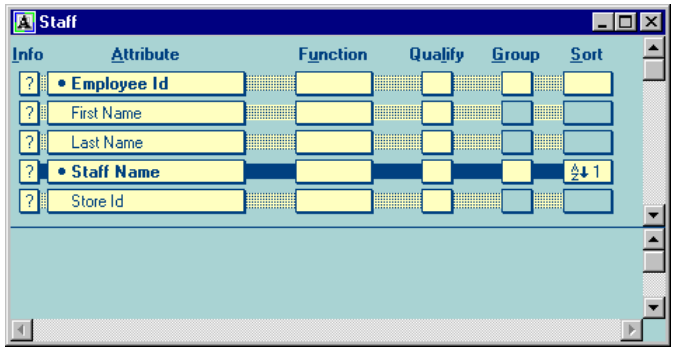
For more information on alias data values, see the Help for your BI Query application.

You can create a data values file to alias data so that a user can see values in the data values list that are meaningful (such as last names), while still using values from the database (such as ID numbers).

To create a data values file that aliases data:

1 In the attribute window as you construct a new query, select the attribute that is to be sent to the database and the attribute you want to serve as the alias (whose values are displayed in the data values list).

2 Click the Sort box corresponding to the alias attribute.



3 Do one of the following:

- Save the query as a data values query file, naming it appropriately in the Save Query dialog box.
- Submit the query and save the results as a data values results file.

4 Name the file after the first attribute (the attribute sent to the database).

Chapter 6

Using Prompts

About Prompts **91**
Types of Prompts 92
Naming Conventions for Prompts 94
Data Types for Prompts 94
Required Versus Optional Prompts 96
Prompts for BI Web Reports 97

Common Tasks Using Prompts **99**
Creating a Single Prompt 100
Inserting a Prompt in a Query 102
Creating a Group Prompt 103
Testing a Group Prompt 103
Creating a Conditional Prompt 105
Testing a Conditional Prompt 105
Importing Prompts 106
Showing Prompt Values in Queries 107

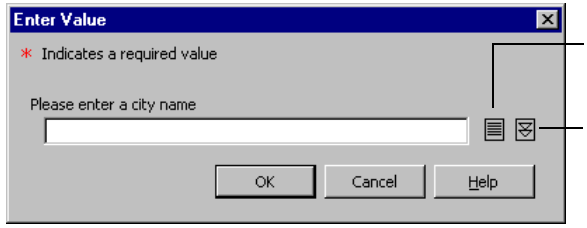
Specifying Web-Reporting Values **107**

About Prompts

A prompt is a request to enter a value or set of values. Typically, you can insert a prompt into a query qualification instead of specifying data values. Each time the user submits the query, the query can retrieve information for different values, depending on what response the user makes to the prompt. You can also include prompts in other contexts, including join conditions and object qualifiers.

A prompt consists of a name and a prompt string. The name distinguishes the prompt from others you may create (and may indicate if it belongs to a special aggregate known as a group prompt). The prompt string is the text that appears in the Enter Value dialog box when users run a query involving the prompt (for example, "Please enter a city name").

The name of the Enter Value dialog box changes depending on the type of prompt in use. For more information, see "Naming Conventions for Prompts" on page 94.


Enter Value
* Indicates a required value
Please enter a city name
OK
Cancel
Help
List Button—Lists previously entered values for the prompt.
Data Values Button—Lists available data values and data sources (such as variables) for the prompt.

You can also associate a particular data type with a prompt and specify whether users must supply a value for the prompt. If the prompt appears in a report for BI Web users, you can specify the appearance of the prompt and, in some cases, an optional list of allowed data values.

BI Query stores prompts in a single `Prompts` file in the `Queries` folder. BI Query does not save the prompts created for a given data model until you close the data model.

> **Note:** If you insert a prompt into a query and then convert that query into a query data object (QDO), all queries using that QDO will require users to respond to the prompt, although the prompt will not be visible in the data object or the SQL for the query. For more information on QDOs, see the *BI Query Data Models User's Guide*.

Types of Prompts

There are three types of prompts: single prompts, group prompts, and conditional prompts.

Single Prompts

A single prompt retrieves a value (or set of values) for a given query. Single prompts usually qualify a single attribute. A single prompt appears in the Enter Value dialog box.

Group Prompts

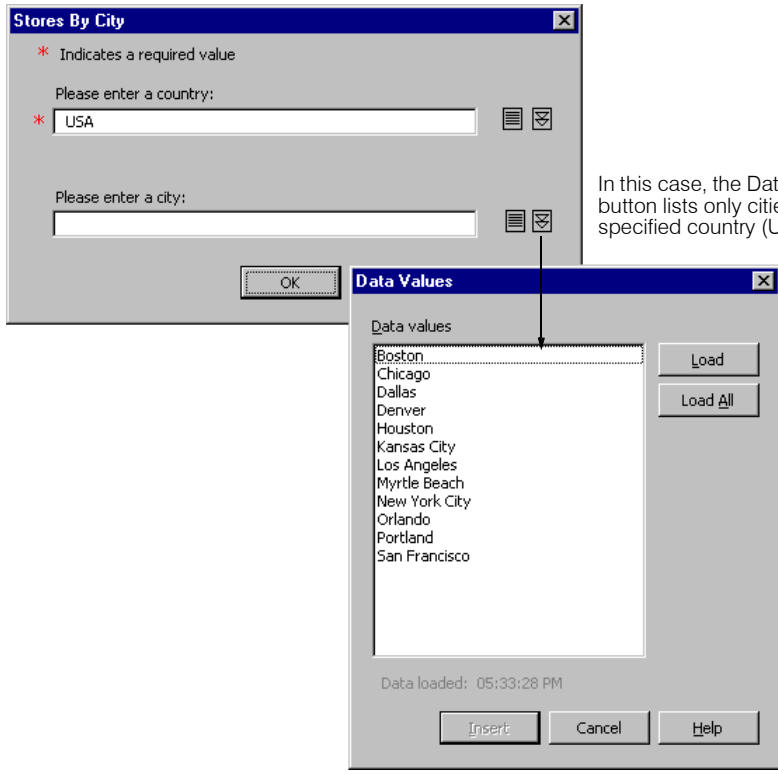
A group prompt is made up of two or more distinct prompts that are grouped by their names. When you run a query that involves multiple prompts in a given group, they all appear in the same dialog box. For example, a group prompt may consist of prompts for the attributes Country and City; a query that qualifies those attributes prompts you to select a country and a city using the same dialog box.

A group prompt can display up to five prompts in one dialog box. BI Query presents the distinct prompts in a group in alphanumerical order. If a query contains more than five prompts in a group, the first dialog box displays the first five prompts (in order), a second displays the next five, and so on. A query can contain more than one group prompt.

Using a group prompt in a query lets users see all the prompts at once and reduces the number of steps in the query. A group prompt also relates one prompt to another.

Conditional Prompts

A conditional prompt is a group prompt in which the available data values for each successive prompt in the group depend on the values you specify for the preceding prompts. For example, a conditional prompt involving Country and City attributes may list only data values for those cities located in the country that you specify.



Conditional prompts can eliminate inappropriate data values and make it easier for users to run a given query. Conditional prompts are also useful in a data model that doesn't let users build their own queries, because they provide a means of drilling down in the data to obtain increasingly detailed information.

Naming Conventions for Prompts

Single prompts can have any name; prompts that qualify an attribute are usually named after the attribute. (For example, if you create a prompt that qualifies the City attribute, the prompt can be named City.) Group prompts (including conditional prompts) have a special naming convention.

Group Prompt Names

The name for each prompt in a group takes the following form:

group_name!*prompt_name*

where *group_name* is the name of the group and *prompt_name* is the name of the prompt in that group. For example, the city prompt in the stores group has the name stores!city. *group_name* is optional; it lets you create several groups of prompts.

Group prompts can also have a special Title prompt associated with them. This prompt stores the title of the group prompt dialog box. If you do not supply a Title prompt, the dialog box for the group is called Enter Values. Title prompt names take the following form:

group_name!!Title

where *group_name* is the name of the group and !!Title is the literal string as written (case is important). The prompt string for a Title prompt represents the title for its group. For example, if you create a Title prompt for the stores group (stores!!Title) and specify its prompt string as "Stores By City", the dialog box that opens for that group is called Stores By City.

Data Types for Prompts

Prompts that qualify attributes automatically use the data type of those attributes. However, if you are creating a prompt that does not qualify an attribute, such as a prompt in a join or object qualifier, you need to set the data type of the prompt. When users respond to the prompt, the value they enter must conform to the data type of that prompt. The data type for a prompt ensures that users enter values that are acceptable to the database.

BI Query groups prompt data types into general categories. When you create a prompt, you select the category that corresponds to the data type that the database expects, rather than the specific type itself.

Example

You want to use a prompt as an argument in a DBMS function that requires Smallint values. When you create the prompt, you select the prompt data type category that corresponds to Smallint data type (in this case, Integer). When the users run a query involving that prompt, they must supply integer values; if they enter character data or floating-point values, BI Query displays an alert box to warn them of the problem, then lets them either modify the value to the correct format or enclose the value in curly braces ({}) to send it to the database as is.

Data Type Categories

You can set the data type for a prompt using the Data Type list in the Edit Prompt dialog box. If you are creating a prompt that qualifies an attribute, the prompt automatically uses the data type of the attribute, even if you assign a different data type to the prompt.

The following table lists the data type categories for prompts:

Data Type Category	Description
(Automatic Formatting)	Ensures that a prompt has the same data type as the attribute it qualifies. Do not select this type for any of the following: • prompts that do not qualify an attribute • prompts that qualify a date, time, or datetime attribute in a Teradata DBMS
Text	Includes the following DBMS data types: c, char, character, long, long varchar, test, varchar, varchar2, vchar, varcharacter.
Integer	Includes the following DBMS data types: integer, smallint, number(p), int, smallint, tinyint, bit, serial, integer1, integer2, integer4, byteint, smint, decimal(p).
Real	Includes the following DBMS data types: dec, decimal(p,s), numeric(p,s), float, real, double precision, largeint, number, number(p,s), money, smallmoney, smallfloat, float4, float8.

Data Type Category	Description
Date	Includes the following DBMS data types: date, datetime (for example, year to day).
Time	Includes the following DBMS data types: time, datetime (for example, hour to second).
Date and Time	Includes the following DBMS data types: date, datetime, timestamp, smalldatetime.
(No Formatting)	Lets users enter any type of data. This is equivalent to placing curly braces ({}) around the prompt name in the qualification tree.

Required Versus Optional Prompts

When you create a prompt, you can specify whether it is required or optional.

Required Prompts Are useful to prevent queries from returning too much data. A required prompt forces users to supply at least one value for the prompt. If users leave a required prompt empty, the query does not run. BI Query marks required prompts with a red asterisk in the prompt dialog box.

Warning! Users can remove any prompt (required or otherwise) from the qualification tree for a query. Therefore, do not use required prompts to control user access to the database.

Optional Prompts Are useful if a query contains different prompts for different users, not all of whom want to use each prompt. Users can choose to enter no values for an optional prompt. In this case, if the prompt qualifies an attribute, the prompt has no effect on the query; BI Query removes the qualification involving the prompt from the `WHERE` clause of the query. However, if the prompt does not qualify an attribute (for example, if it is an argument to a function), BI Query does not remove the prompt; in this case, an empty prompt value may produce a DBMS syntax error.

> **Note:** Each prompt retains the last value selected by you and displays that value as the default the next time it opens. If you leave an optional prompt empty, you effectively erase the last value.

Prompts for BI Web Reports

For BI Web users who access published standard reports or BI Query reports, you can specify allowed data values for prompts in the corresponding queries. These values are known as web-reporting values. When BI Web users open and refresh a report whose query has a prompt, they can select a value for the prompt from the web-reporting values you have specified.

For more information, see "Creating Data Values Query Files" on page 85.

As an alternative to specific values, you can also make available the data values from a data values query file, if one exists for the prompt or the attribute (if the prompt qualifies an attribute).

> **Note:** If data values query files exist for both the prompt and the attribute, only the file associated with the prompt supplies data values.

Prompt Display Types for Reports in BI Web

When you create a prompt for a BI Web report, you must design its interface. For a given prompt, you can select one of the following display objects:

Object	Description
	Text Box—lets the user type one or more values. This is the default object for web-reporting values. You can use a data values query file for web-reporting values only if the display object is a text box. If there is no data values query file associated with the prompt or the attribute it qualifies, then the prompt uses the data values from the default `SELECT DISTINCT` query for the attribute.
	List Box—lets the user select one or more values from a list.
	Combo Box—lets the user type a value or select one from a list.
	Option Buttons—let the user select exactly one value from a group. If you use option buttons for a prompt, it must be a required prompt.
	Check Boxes—let the user select multiple values from a group.
	Date Entry—lets the user enter a date value in the form `YYYY-MM-DD`.
	Time Entry—lets the user enter a time value in the form `hh:mm AM`.
	Date & Time Entry—lets the user enter a date-and-time value in the form `YYYY-MM-DD hh:mm AM`.

The display object that you specify for a prompt determines how the BI Web user's browser displays the web-reporting values. For example, if you specify 7 web-reporting values and select check boxes as the display object, the BI Web user can select the prompt value(s) from among 7 check boxes.

The display object also determines how many values the user can select. If in the previous example you had selected option buttons as the display object, the BI Web user could select only a single value for the prompt. You can also override the properties of the display object and force users to select a single value.

Note: If the display object is a text box, date entry, time entry, or date & time entry, you cannot specify particular web-reporting values. You must specify these values for all other display objects.

The following table summarizes the available prompt options based on the display object:

Object Type	Required or Optional Prompt?	Multiple Selections Allowed?	Data Values Query File Allowed?	Specific Data Values Allowed?
Text box	Either	Yes	Yes	No
List box	Either	Yes	No	Yes
Combo box	Either	No	No	Yes
Option buttons	Required	No	No	Yes
Check boxes	Either	Yes	No	Yes
Date entry	Either	No	No	No
Time entry	Either	No	No	No
Date & time entry	Either	No	No	No

Common Tasks Using Prompts

A typical application of a prompt is to improve the scope of a given query. For example, you could create a query about customer payment information that prompts you for a country name each time you submit the query. When you want information on customers located in Argentina, you submit the query and type `Argentina` in response to the prompt; when you want information on customers located in Australia, you submit the query and type `Australia` in response to the prompt, and so on.

For tables that contain a large number of rows, you can use prompts to improve query performance and prevent users from submitting queries that retrieve more data than they need. For example, regional sales managers may need sales information only for their regions. When they submit queries against the table containing sales information, the queries prompt them to specify the regions they want.

Other Prompt Applications

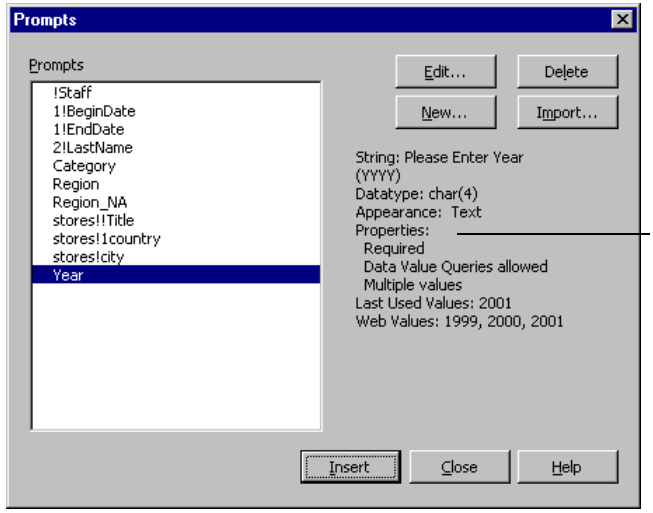
For information on using prompts in join conditions and object qualifiers, see the *BI Query Data Models User's Guide*.

In addition to using prompts to qualify attributes, you can include a prompt in any context where you want the user to supply a value. For more information on prompts and their applications, see the following topics:

- "Using Functions, Prompts, and Other Values in Variables" on page 116
- "Adding Calculated Columns" on page 160
- "Applying a Function Before Grouping" on page 184

Creating a Single Prompt

Use the Prompts and Edit Prompt dialog boxes to create a prompt and specify its properties.



The Prompts dialog box lists the properties of the selected prompt.

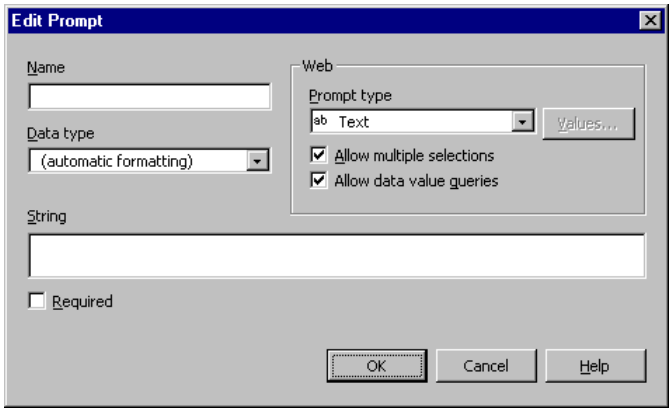
If the list is too long to view, click in the list, then use the up and down arrow keys to scroll through the items in the list.

For information on modifying and deleting prompts, see the Help for your BI Query application.

To create a single prompt:

1 On the Edit menu, click Prompts. The Prompts dialog box opens.

2 In the Prompts dialog box, click New. The Edit Prompt dialog box opens.



3 In the Edit Prompt dialog box, type the name of the prompt into the Name box.

4 In the Data Type list box, select the data type for the prompt.

5 In the String box, type the phrase that you want to prompt the user (for example, "`Please enter an employee ID`").

6 To force users to supply a value for the prompt, select Required; to make the prompt optional, clear Required.

7 If the prompt will appear in a published standard report or BI Query report for BI Web users, specify its web properties in the Web area. These properties apply only to prompts that appear in reports in the BI Web Portfolio. Do the following:

   a) From the Prompt Type drop-down list, select the object for displaying the allowed data values.

   b) To let BI Web users enter multiple values for the prompt, select Allow Multiple Selections. This check box is enabled only if you have selected Text Box, List Box, or Check Boxes from the Prompt Type drop-down list.

For more information, see "Naming and Storing Data Values Files" on page 80.

c) To make data values from a data values query file available to the user, select Allow Data Value Queries. This check box is enabled only if you have selected Text Box from the Prompt Type drop-down list. (You must create the file if it does not exist or does not have the correct name.)

For more information, see "Specifying Web-Reporting Values" on page 107.

d) To specify allowed data values explicitly, click Values. The Web Reporting Values for Prompt dialog box opens. Specify the values you want to make available to users, then click OK. (You cannot specify values if you have selected Text Box, Date Entry, Time Entry, or Date & Time Entry from the Prompt Type drop-down list.)

8 Click OK, then click Close to close the Prompts dialog box.

Inserting a Prompt in a Query

You define a qualification with a prompt in the same way you would a qualification with a value. If you qualify more than one attribute in a query with the same single prompt, the value you specify for the prompt applies to each attribute.

To insert a prompt when forming a qualification:

1 In an attribute window, click the Qualify box for an attribute.

2 In the qualification tree, place the cursor in the Qualification box for the attribute.

3 Click the Data Values button, then select Prompts from the list.

4 In the Prompts dialog box, select a prompt. If the prompt you want isn't listed, you can create it.

5 Click Insert.

6 If your query is complete, click the Run Query button on the Application toolbar, or save it for later submission.

Note: Curly braces ({}) around a prompt value let the database receive formats that BI Query doesn't support. If you place curly braces around the prompt name in the qualification tree, then when the query runs and the user enters values as prompted, each value is enclosed in curly braces, making it unnecessary for the user to remember to add them.

Creating a Group Prompt

To make a group prompt, you create two or more prompts and name them with the group prompt naming convention.

To create a group prompt:

For more information, see "Naming Conventions for Prompts" on page 94.

1 Create the prompts you want. You will require at least two. Name each using the group prompt naming convention (for example, `stores!city`, `stores!country`).

   The prompts are listed in alphanumeric order in the prompt dialog box (`stores!city` before `stores!country`, but `stores!1country` before `stores!city`), so name them according to the order you want them to have.

2 To specify a title for the group prompt dialog box, create a Title prompt with the name *`group_name`*`!!Title`, where *`group_name`* is the name of the group from step 1 and `!!Title` is the literal string as written (case is important). Type the title in the Prompt String box.

Note: If you do not create a Title prompt for a group, the dialog box for the group prompt is called Enter Values.

Testing a Group Prompt

Before distributing a group prompt, you should test it to ensure that it works properly. When you submit the query, the group prompt dialog box displays the prompts in the specified order and with the specified title.

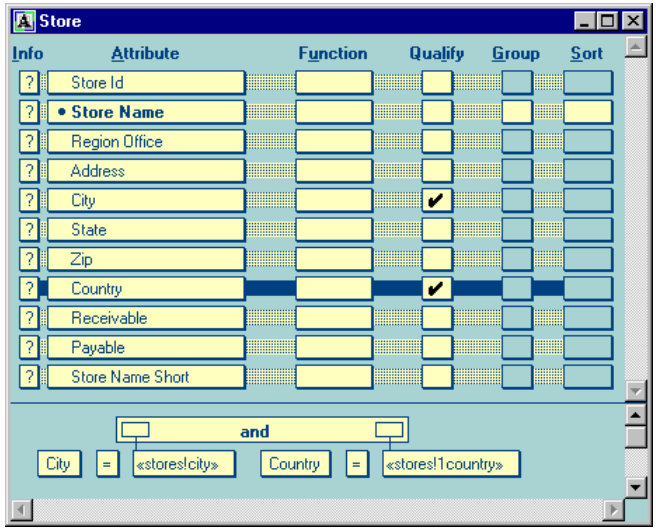
To test a group prompt:

1 In an attribute window, select the attributes you want.

2 For each attribute, do the following:

a) Select the Qualify box for the attribute.

b) On the Edit menu click Prompts, then specify one of the prompts that have been grouped with the naming convention from the list in the Prompt dialog box.



3 Submit the query.

4 In the group prompt dialog box, specify a value in the first box. (Type it or use the Data Values button to select a value from a list.)

5 Specify values in the subsequent boxes.

6 Click OK.

Creating a Conditional Prompt

For more information on data values query files, see "Creating Data Values Query Files" on page 85.

A conditional prompt is a special kind of group prompt that uses a data values query to restrict the available data values for successive prompts in the group. The second prompt in a conditional prompt uses a data values query that includes the first prompt. In this way, the values supplied by the first prompt qualify the query that supplies a choice of values for the second prompt.

To create a conditional prompt:

1. Create the group prompts you want. Use the group prompt naming convention so that BI Query presents the prompts in the order you want.
2. To restrict data values for the first prompt in the group, create and save a data values query file that returns the data values you want for the first prompt. Name this file after the name of the first prompt. When you insert this prompt into a qualification, BI Query makes the results of the query file available in the corresponding prompt dialog box.

   If you do not create a data values file for the prompt, BI Query uses the default `SELECT DISTINCT` query to generate the available data values.
3. Create a new query to return the data values you want for the second prompt. In this query, use the first prompt to qualify the attribute that corresponds to the first prompt, then apply the appropriate operator.
4. Save the query as a data values query file using the name of the second prompt.

Testing a Conditional Prompt

Before using or distributing a conditional prompt, you should test it to ensure that it works properly.

To test a conditional prompt:

1. In an attribute window, click the attributes you want.
2. Use the first prompt to qualify the corresponding attribute.
3. Use the second prompt to qualify the corresponding attribute.

4 Submit the query.

5 To enter a value for the first prompt, do one of the following:

- In the group prompt dialog box, type a value into the first box.
- If data values haven't been disabled for the attribute, click the Data Values button and select a value from the list.

6 Click the Data Values button for the second prompt. From the pop-up menu that opens, click Data Values.

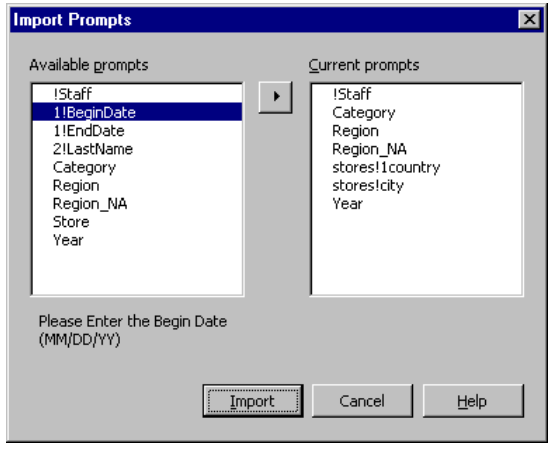
7 In the Data Values dialog box, select the value(s) you want. (Depending on the operator you applied to the qualification, you can select one or more values.)

8 Click Insert.

9 In the group prompt dialog box, click OK.

Importing Prompts

To use prompts created by someone else, import them from that person's Prompts file (stored in that person's Queries folder). Use the Import Prompts dialog box.



To import prompts:

1 On the Edit menu, click Prompts.

2 In the Prompts dialog box, click Import.

3 In the Select Import File dialog box, select the file that contains the prompts you want to import, then click Open. The Import Prompts dialog box opens.

4 Select the prompts you want under Available Prompts, then click the right arrow button to move the selected prompts to the Current Prompts box.

5 Click Import, and in the Prompts dialog box, click Close.

Showing Prompt Values in Queries

Values that have been typed into prompts at run time can be viewed in the SQL string area of the Query window. To view the SQL string, the Show SQL String preference must be set.

To show a prompt value in a query:

1 Open the Query window.

2 In the Query window, select Show Prompt Values.

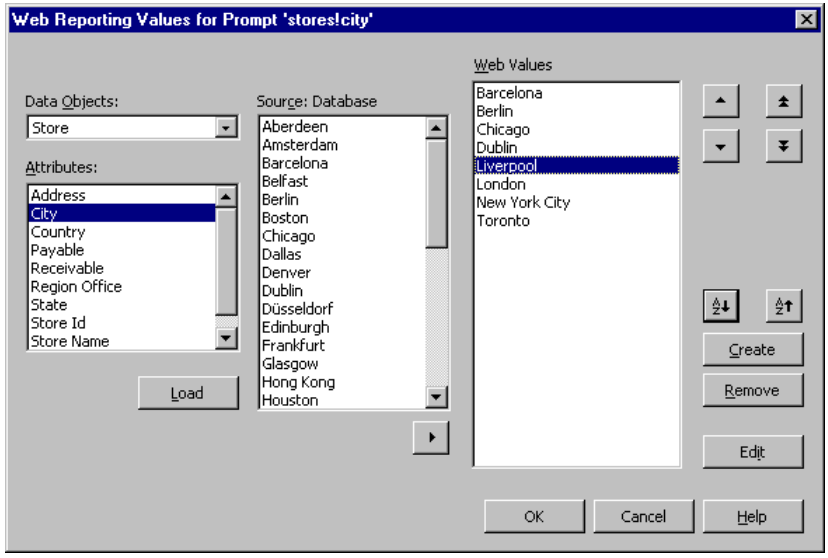
If you have entered prompt values, they are usually displayed in the WHERE clause of the SQL string. (For calculated attributes, they appear in the SELECT clause.) If no values were entered for the prompt, the prompt does not appear in the SQL string.

Specifying Web-Reporting Values

If you are creating a prompt for a BI Web report and the prompt does not use a text box, date entry, time entry, or date & time entry, you can specify the web-reporting values explicitly. The values you specify can come from the database (in which case, you must specify a particular data object and attribute in that object). You can also enter your own values directly.

To specify web-reporting values for a prompt:

1 Open the Web Reporting Values for Prompt dialog box for the prompt. (In the Edit Prompt dialog box, click Values).



2 To select values from the database, do the following until all the values you want appear in the list:

a) From the Data Objects list, select an object. From the Attributes list, select an attribute.

b) Click Load. The Source: Database list displays all distinct values in the database for the selected table and attribute.

c) From the Source: Database list, select the web-reporting values that you want to use for the prompt, then click the right arrow button. The selected values appear in the Web Values list.

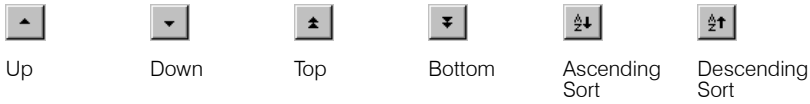
3 To specify your own values, do the following until all the values you want appear in the list:

a) Click Create. The New Allowed Data Value dialog box opens.

b) In the Data Value box, type a data value. Click OK.

For more information on sorting and editing web-reporting values, see the Help for your BI Query application.

4 Set the order for the values. The top-to-bottom order of the values in the Web Values list is the order in which they appear in the BI Web prompt form. Use the available buttons:



Up | Down | Top | Bottom | Ascending Sort | Descending Sort

5 Click OK.

Chapter 7

Using Variables

About Variables **113**
Typical Uses for Variables 114

Expressions for Variables **114**
Using Literals in Variables 114
Using Placeholders in Variables 115
Using Functions, Prompts, and Other Values in Variables 116

Creating and Using Variables **117**
Using Variables to Qualify an Attribute 119
Importing Variables from a File 120
Using Variables to Supply User or Group Names 121

About Variables

A variable (or automatic variable) is a predefined operation that you can use to qualify a query. Typically, variables qualify attributes, but you can also use them in object qualifiers. Using variables can simplify potentially complex queries.

A variable is defined by an expression. When you submit a query that includes a variable, BI Query evaluates the corresponding expression and adds it to the `WHERE` clause for the query.

A variable can contain attributes, operators, values in the database, calculations, functions, subqueries, prompts, and other variables (anything that's valid in an SQL `WHERE` clause). You can also use special placeholders in the expression for a variable to increase its scope.

Variables are stored in the `Prompts` file in the `Queries` folder and therefore must have names that are different from those given to prompts. You can import variables stored in another user's `Prompts` file.

Predefined Variables

BI Query includes three predefined variables:

DBUserID Supplies the database user name for the user currently logged on to the database.

BIUserID Supplies the BI Server user name for the user currently logged on to BI Server.

BIGroupID Supplies the list of groups to which the user currently logged on to BI Server belongs. You can use this variable only if the BI Server administrator has created BI Server groups and associated users with those groups.

You cannot edit the predefined variables.

> **Note:** If a query using a variable is converted into a query data object, the variable is preserved within the object. For more information on query data objects, see the *BI Query Data Models User's Guide*.

Typical Uses for Variables

Variables can carry out a variety of operations. In their simplest form, they can retrieve values in the database so you don't have to specify them in queries. For example, when you need information about retailers in a particular country, you can use a variable to determine the states or provinces that are included in that country, rather than specify each state or province yourself.

Variables can also perform calculations based on database functions to work out date, time, and other values. For example, you can use a variable to retrieve retailer sales that were placed today, before today, and so on. The variable supplies what today's date is, and BI Query returns the corresponding results.

Variables are especially useful for queries that are run on a regular basis or are scheduled to run at specific intervals because they can automatically supply data values for the time when they are run.

Expressions for Variables

When you submit a query qualified with a variable, BI Query evaluates the expression for the variable and adds the result to the `WHERE` clause for the query. Each variable usually contains information about what attribute to qualify, what operator to use, and what value to test against the attribute.

There are two ways to provide attribute and operator information in a variable:

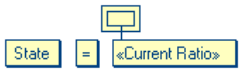
- Use an actual attribute name and operator.
- Use placeholders for attributes and operators.

Using Literals in Variables

A user-defined variable that names an actual attribute and operator always qualifies that attribute using that operator, even if you insert it into the qualification for a different attribute in the query. Unless you define the variable with placeholders, BI Query removes the qualified attribute and qualification operator from the `WHERE` clause for the query.

Example

A query includes the Store Name and State attributes of the Store data object. The State attribute is qualified with the user-defined variable Current Ratio.



The expression for Current Ratio is

```
Store.Receivable > Store.Payable
```

The syntax for variable expressions depends on the DBMS you're using; the examples provided here are based on ODBC.

When you run the query, BI Query substitutes the variable for the qualification of the Store Name attribute. In other words, the original qualification,

```
Store.State = «Current Ratio»
```

becomes

```
Store.Receivable > Store.Payable
```

Using Placeholders in Variables

As an alternative to actually specifying attribute and operator information in a variable, you can create a variable that uses placeholders for the attribute and operator. The placeholders let you create variables that return results based on the particular attribute qualified and the operator used in the qualification. There are two placeholders for variables:

^Attribute^ The placeholder for the qualified attribute. When you submit a query containing the ^Attribute^ placeholder, BI Query replaces the placeholder with the qualified attribute; this attribute becomes part of the `WHERE` clause for the query.

^Operator^ The placeholder for the operator in the qualification. When you submit a query containing the ^Operator^ placeholder, BI Query replaces the placeholder with the operator from the qualification; this operator becomes part of the `WHERE` clause for the query.

Use the ^ Attribute ^ and ^ Operator ^ buttons in the Edit Variable dialog box to insert placeholders in the expression for a variable.

> **Note:** The predefined variables, DBUserID, BIUserID, and BIGroupID, use placeholders for attributes and operators. This means you can directly qualify any attribute using these variables.

Example

The variable Exceeding Credit is defined as follows:

```
^Attribute^ ^Operator^ Retailers.Credit
```

If you qualify the attribute Receivable in the Retailers data object with this variable,

```
Retailers.Receivable > «Exceeding Credit»
```

then the actual qualification becomes

```
Retailers.Receivable > Retailers.Credit
```

If you qualify the Invoice_amount attribute of the Retailers object with the same variable,

```
Retailers.Invoice_amount < «Exceeding Credit»
```

then the query returns the list of retailers who have been invoiced for amounts less than their credit limit:

```
Retailers.Invoice_amount < Retailers.credit
```

Using Functions, Prompts, and Other Values in Variables

Typically, the expression for a variable takes the form of a comparison between an attribute and a value. An operator specifies the nature of the comparison (for example, `Sales.order_id = '10002'`). The expression can also include functions, prompts, or other variables. These additional terms often supply the comparison value for the attribute.

For more information on functions, see "Using Functions in Query Expressions" on page 189.

Functions Use a function in a variable to create an expression that performs a calculation. For example, you can include in an expression the DBMS function for the current date:

```
^Attribute^ ^Operator^ {fn now()} - {fn dayofmonth({fn now()})}
```

This variable determines the date for the beginning of the month. You can use this variable to find all retailer sales received since that date.

Prompts Use a prompt in a variable if you want to qualify the query but do not want to restrict the comparison to a specific value. For example, in the following expression,

```
{fn week(^Attribute^)} ^Operator^ «weekofyear!1week» AND
{fn year(^Attribute^)} ^Operator^ «weekofyear!2year»
```

the variable includes the weekofyear group prompt. This variable returns results based on the week and year that you specify.

Other Variables Use other variables in a variable as the building blocks for a more complex expression. For example, the following expression,

```
^Attribute^ ^Operator^ «Today» - 1
```

includes the Today variable, which determines today's date. The resulting expression returns yesterday's date.

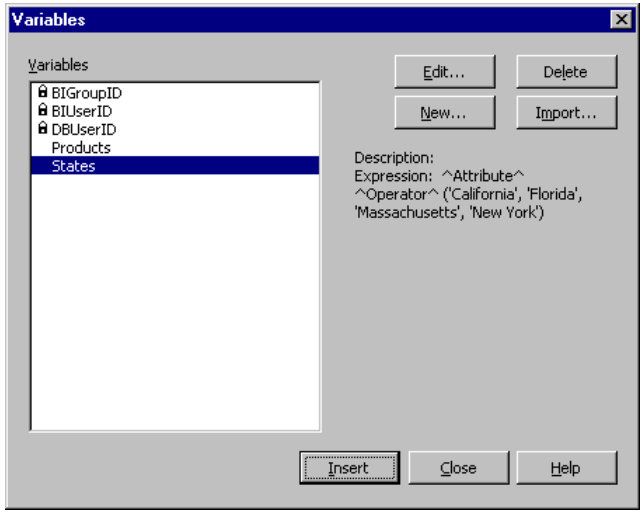
Creating and Using Variables

For more information, see "Creating Calculated Attributes" on page 186.

Before you create the expression for a variable, you can test that it returns the results you want by creating a calculated attribute using the same expression. Use the attribute in a query, then verify the results. If the expression is correct, you can use the Edit Variable dialog box to create the variable.

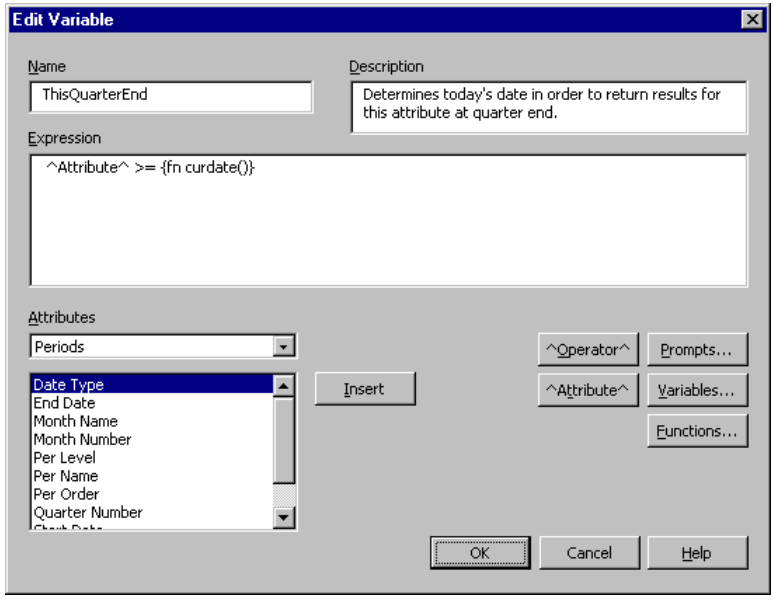
To create or edit a variable:

1 On the Edit menu, click Variables. The Variables dialog box opens.



You can't edit any variables with lock icons beside them.

2 Click New to create a new variable or, to edit an existing variable, select the variable, then click Edit. The Edit Variable dialog box opens.



3 Type or edit the name and description of the variable in the Name and Description boxes.

4 In the Expression box, type the expression that supplies the value for the variable. (You can include database values and functions, calculations, subqueries, other variables, and prompts.)

Use curly braces ({}) to enclose functions or formats supported by your DBMS (for example, a current date function) that BI Query doesn't support.

You can also use the available buttons to build the expression:

- To insert an attribute into the expression, select a data object from the Attributes drop-down list, then select the attribute from the list below. Click Insert.
- To specify a placeholder for an attribute, place the cursor in the Expression box, then click ^ Attribute ^.
- To specify a placeholder for an operator, place the cursor in the Expression box, then click ^ Operator ^.
- To insert a prompt, click Prompts, then use the Prompts dialog box to create or select the prompt and insert it into the expression.
- To insert a variable, click Variables, then use the Insert Variable dialog box to select the variable and insert it into the expression.
- To insert a function, click Functions, then use the Functions dialog box to create or select the prompt and insert it into the expression.

For more information on prompts, see "About Prompts" on page 91.

5 Click OK. In the Variables dialog box, click Close.

Using Variables to Qualify an Attribute

You can qualify an attribute so that a variable specifies the search conditions for the query. A variable can contain anything that's valid in an SQL WHERE clause (attributes, operators, values in the database, calculations, functions, and subqueries). Variables are especially useful for queries that run on a regular basis or are scheduled to run at specific intervals using BI Server's scheduling facility.

For more information on scheduling, see the *BI Server Scheduling Guide*.

To qualify an attribute with a variable:

1 In the qualification tree, place the cursor in the Qualification box for the attribute.

2 On the Edit menu, click Variables.

3 In the Variables dialog box, select the variable you want (or create or edit one.

4 Click Insert.

Importing Variables from a File

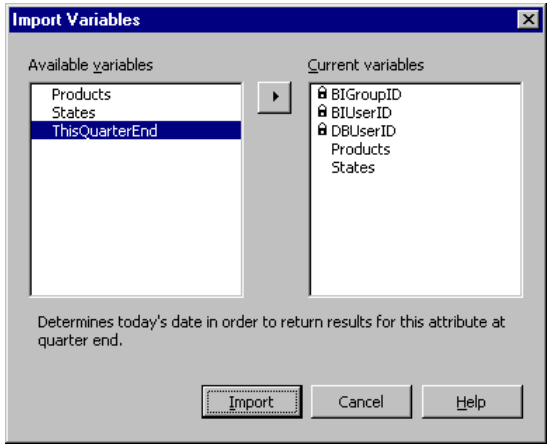
When you want to use variables created by another user, you can import them from that user's `Prompts` file.

To import variables from a file:

1 On the Edit menu, click Variables.

2 In the Variables dialog box, click Import.

3 In the Select Import File dialog box, open the `Prompts` file containing the variables you want to add to the current data model. The Import Variables dialog box opens.



4 Select the variable(s) you want from the Available Variables list, then click the right arrow button to move the selected variables to the Current Variables list.

5 Click Import, then in the Variables dialog box, click Close.

Using Variables to Supply User or Group Names

For more information on security tables, see the *BI Query Data Models User's Guide*.

The creator of the data model can include a security table—a data object that associates data with the value(s) returned by one of the predefined variables (DBUserID, BIUserID, or BIGroupID). If such an object exists in the data model, you can use the predefined variables to qualify a query so that it retrieves results customized for you. If you distribute that query to other users, it retrieves results customized for each of them. You can also use these variables to restrict data to certain users or groups.

To qualify a query using a user or group name variable:

1 Double-click the data object that serves as the security table.

2 In the attribute window, click the Qualify box for the attribute you want to qualify. This attribute must contain data that corresponds to the data returned by the variable you want to use. (For example, BIUserID returns the BI Server user name for the current user. The attribute that you qualify with this variable must store BI Server user names.)

3 In the qualification tree, place the cursor in the Qualification box.

4 Click the Data Values button, then select Variables from the list.

5 In the Variables dialog box, select the appropriate variable (DBUserID, BIUserID, or BIGroupID).

6 Click Insert.

> **Warning!** Users must be logged on to BI Server when submitting queries that execute the BIUserID or BIGroupID variables. Otherwise, the variable can't supply the user or group name(s). This is true even if the user is submitting a local copy of the query that would not otherwise require logging on to BI Server.

Chapter 8

Working with Results

About Results **125**
Opening Results 126

Modifying a Query for Selected Results **127**

Using Super Queries **129**
About the Super Query Window 129
Opening the Super Query Window 132
Guidelines for Editing Super Queries 132
Editing Component Queries 133
Editing an Operation in the Super Query Window 134
Correcting Errors 135

Filtering Results **136**
Getting Results in Sorted Order 136
Reordering Columns 137
Narrowing the Range of Results 139

Combining Results Sets **140**
About Appended Columns 141
Combining Results by Appending Rows 142
About Column Joins 143
Types of Column Joins 144
Combining Results by Joining Columns 146

About Results

For more information on saving results as text, see "Exporting Results to a Text File" on page 226.

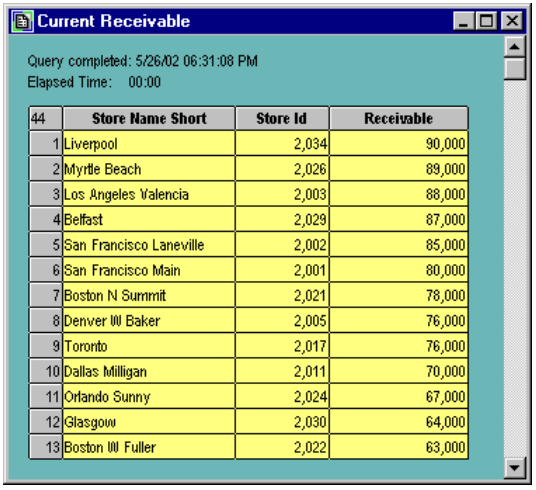
Results are the information returned from your database after you submit a query. Query results can be used "as is" or as a source of data for a report. Results can be returned to a results window on your computer screen, where they can be combined, filtered, and formatted, as well as stored and distributed, without further database access. You can also submit a query so that the results are stored in a text file or in the database, then load the results at a later time.

If your querying environment includes BI Server and you have the necessary system permissions, you can schedule a query to run automatically. In that case, the results are returned to the BI Server Repository, where they can be retrieved by you as well as by other users with appropriate permissions. All the results returned from a query, taken together, are called a results set.

Spreadsheet and Form Views

You can search for specific content in query results. For more information, see the *BI Query Data Models User's Guide*.

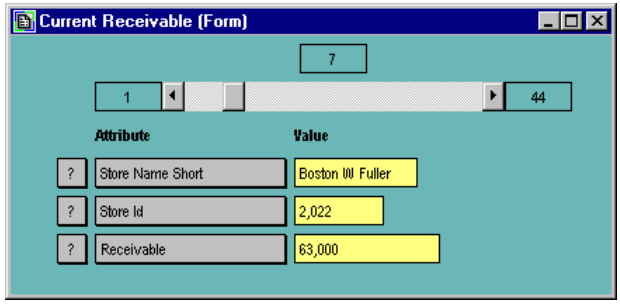
By default, results are returned in a spreadsheet view in the results window. It displays multiple records (each record is displayed as a row in the spreadsheet). The columns reflect the attributes selected in the query. The name of the results window corresponds to the name of the query.



44	Store Name Short	Store Id	Receivable
1	Liverpool	2,034	90,000
2	Myrtle Beach	2,026	89,000
3	Los Angeles Valencia	2,003	88,000
4	Belfast	2,029	87,000
5	San Francisco Laneville	2,002	85,000
6	San Francisco Main	2,001	80,000
7	Boston N Summit	2,021	78,000
8	Denver W Baker	2,005	76,000
9	Toronto	2,017	76,000
10	Dallas Milligan	2,011	70,000
11	Orlando Sunny	2,024	67,000
12	Glasgow	2,030	64,000
13	Boston W Fuller	2,022	63,000

The results window for the Current Receivable query

You can also display a row in form view by double-clicking the row number in the results window. Form view displays one row at a time and provides a way to navigate through the rows you have retrieved.



Formatting Results

You can format the results that appear in a results window, either by formatting the attributes involved in the query or by formatting the result columns directly. For more information on formatting results, see "About Data Formats" on page 57.

Opening Results

BI Query saves results as two files: one with the extension `.qrd` (ASCII text format for use in other applications) and one with the extension `.qrr` (BI Query-specific information, such as formats for each column and the SQL used to retrieve the results). When you open results in BI Query, only the .qrd file appears in the Open Results File dialog box.

To open a results file:

1 On the Results menu, click Open.

2 In the Open Results File dialog box, double-click the `.qrd` file you want to open.

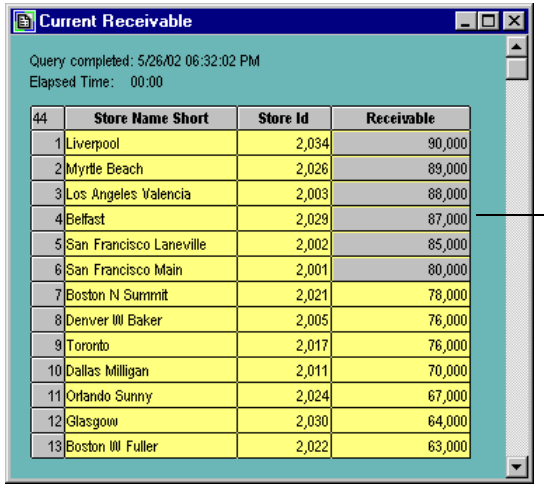
Modifying a Query for Selected Results

If you want to retrieve more detailed data about specific results you've retrieved, you can do so in Ad Hoc Drill-Down Mode. In this mode, you select one or more results cells, then submit a query that returns results only for those values.

To modify a query for selected results:

1 In the results window, on the Results menu, click Ad Hoc Drill Down.

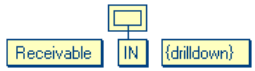
2 Select the cell(s) on which to base your query.


Current Receivable
Query completed: 5/26/02 06:32:02 PM
Elapsed Time: 00:00
44
Store Name Short
Store Id
Receivable
1 Liverpool 2,034 90,000
2 Myrtle Beach 2,026 89,000
3 Los Angeles Valencia 2,003 88,000
4 Belfast 2,029 87,000
5 San Francisco Laneville 2,002 85,000
6 San Francisco Main 2,001 80,000
7 Boston N Summit 2,021 78,000
8 Denver W Baker 2,005 76,000
9 Toronto 2,017 76,000
10 Dallas Milligan 2,011 70,000
11 Orlando Sunny 2,024 67,000
12 Glasgow 2,030 64,000
13 Boston W Fuller 2,022 63,000

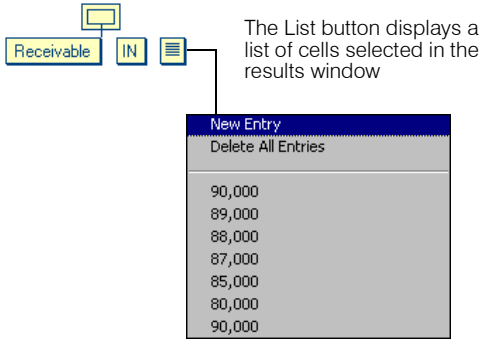
The original query returns a list of store names, store IDs, and receivables.

Cells you select using Ad Hoc Drill Down are indicated in grey.

When you select a single cell, {drilldown} appears in the qualification tree instead of the contents of the cell.

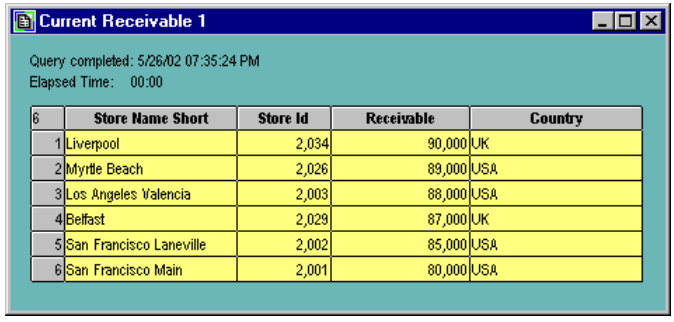


When you select more than one cell, the List button appears; you can use it to view and modify the list of cells you selected.



3 Do one of the following:

- To retrieve results that match these items, resubmit the current query.
- To retrieve different results for these items, modify the current query, then submit it.



6	Store Name Short	Store Id	Receivable	Country
1	Liverpool	2,034	90,000	UK
2	Myrtle Beach	2,026	89,000	USA
3	Los Angeles Valencia	2,003	88,000	USA
4	Belfast	2,029	87,000	UK
5	San Francisco Laneville	2,002	85,000	USA
6	San Francisco Main	2,001	80,000	USA

The Current Receivable query has been modified in Ad Hoc Drill-Down Mode to return store and country information for the stores with the six largest receivable amounts.

4 To work with the same subset of results later, save the new query.

Note:

- SQL `IN` clauses are created in the current query. The query will be resubmitted to retrieve results matching only the items selected.
- To modify the current query to retrieve additional columns of results for these items, display the attribute window(s) for the appropriate items, make the changes you want, then submit the query.

To cancel Ad Hoc Drill-Down Mode:

In the results window, on the Results menu, click Ad Hoc Drill Down.

Using Super Queries

For more information about editing super queries, see "Guidelines for Editing Super Queries" on page 132.

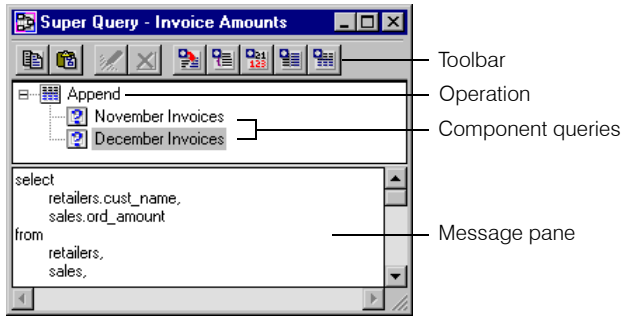
A super query is a query that either returns a results set and then filters it or returns multiple results sets and combines them. You can save, open, and edit these queries in BI Query. However, you can't use them in data values query files and you can't save their SQL as a text file.

In contrast, a standard query returns a single set of results that has not been manipulated in any way. If you save a query before you submit it, or if you save a query after you submit it but without manipulating the results, you are saving it as a standard query. You can save a standard query, open it, edit it, or save it as a data values query file, and you can save the underlying SQL for use in another application.

About the Super Query Window

For more information on combining results, see "Combining Results Sets" on page 140.

Use the Super Query window to combine queries. You can also use it to sort results, reorder columns, and specify a range of results. The Super Query window is divided into two panes separated by a splitter bar, which you can use to resize the panes.



The top pane displays the structure of the query using a tree list. Each component query and each operation is included in the tree and is represented by one of the following icons:

Icon	Description
	Component query
	Sort operation
	Range operation
	Reorder operation
	Join operation
	Append operation
	Error in operation

The operations performed on a component query's results appear above the component query in the tree. For example, if the results of a component query are reordered, then sorted, the tree list shows three branches, first the component query, then the reorder operation, then the sort operation above it.

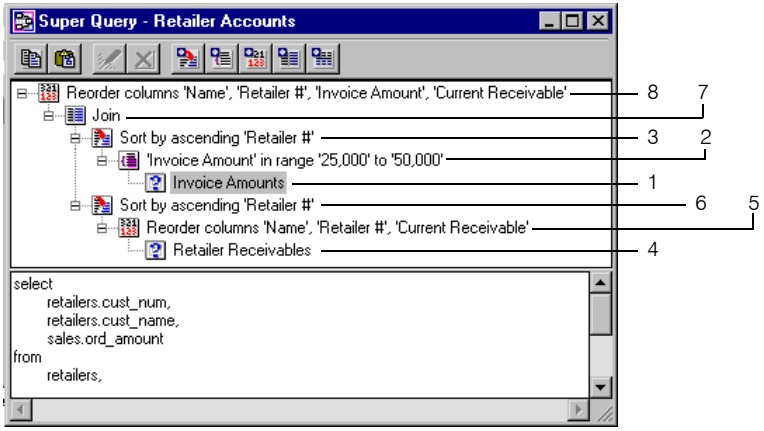
Note: If a join or append operation is performed on two of the component queries, it's shown in the window above the queries and their associated operations. Any operation that appears above a join or append operation is applied to the results of the join or append.

The bottom, message pane displays information about the currently selected item in the query. When you select a component query, the bottom pane displays the query's SQL string. When you select an operation, the bottom pane shows information about what the operation does to the query results. If there's an error in the query, the bottom pane displays information about why the error occurred and how to correct it.

Example

When you're working with complex queries, you can see where a join or append occurs by collapsing the tree structure, then expanding the branches one by one.

The following example shows the Super Query window for the Retailer Accounts query:



This super query performs the following sequence of operations:

1 Retrieves the results of the Invoice Amounts query.

2 Applies a range to the results.

3 Sorts by Retailer #.

4 Retrieves the results of the Retailer Receivables query.

5 Reorders the columns

6 Sorts by Retailer #.

7 Joins the results from step 3 with the results from step 6.

8 Reorders the columns to produce the final query.

Opening the Super Query Window

You can open the Super Query window from an open query or a results set.

To open the Super Query window from a current query:

On the Query menu, point to Super Queries, then click Show Super Query. The new query appears in the Super Query window where you can combine it with saved queries.

To open the Super Query window from a current query results set:

On the Query menu, click Show.

The Super Query window also appears when you open a saved query that was created using the Super Query window.

Guidelines for Editing Super Queries

When you want to change the results a combined query retrieves, you can edit it in the Super Query window. You can change each component query (each query that makes up the larger query) and each operation performed on the results it returns. For example, you can add an attribute to a component query, then change a sort operation to use the new attribute.

When you edit a query in the Super Query window, follow these guidelines:

- Start with the innermost query and work up through the tree. This makes it easier to correct any errors.
- Save your work regularly.
- Don't edit a combined query by modifying the SQL string. Your changes won't be checked for errors.

Editing Component Queries

You edit a component query (a query that's part of a larger query) by copying it from the Super Query window into the data model. The component query then becomes the current query, allowing you to open the attribute windows you want to use and make your changes. When you've finished editing the query, you paste it back into the Super Query window. If any errors occur as a result of making the changes, you need to correct them before you can save and submit the query.

To edit a component query:

1 Do one of the following:

- If the query you want to edit isn't open, on the Query menu, click Open. In the Open Query dialog box, select the query, then click Open.
- With the cursor in the query's results window, on the Query menu, click Show.

2 In the Super Query window, right-click the component query you want to edit, then click Copy Query to Model on the pop-up menu that opens.

3 Minimize the Query window for the component query.

4 In the design window, click the appropriate data object(s) and make your changes to the component query.

5 In the Super Query window, right-click the component query you've edited, then click Paste Query from Model on the pop-up menu that opens.

If you can't find the Super Query window in the list, on the Window menu, click More Windows, then use the Select Window dialog box to open the window.

6 Correct any errors.

7 Save the query.

Editing an Operation in the Super Query Window

You can change how an operation manipulates query results by editing the operation in the Super Query window.

When you select the operation you want to edit, BI Query opens the same dialog box you used to create the operation. (The Join Columns dialog box is slightly different. You can't change the results sets that are being joined.) When you click OK in the dialog box, the changes are applied to the query. If any errors occur as a result of the changes, you need to correct them before you can save or submit the query.

To edit an operation:

1 Do one of the following:

- If the query you want to edit isn't opened, on the Query menu, click Open. In the Open Query dialog box, select the query, then click Open.
- In the query's results window, on the Query menu, click Show.

2 In the Super Query window, right-click an operation, then click Edit Operation on the pop-up menu that opens.

3 In the dialog box that opens, make your changes to the operation. (The type of operation you're editing determines which dialog box appears.)

4 Click OK.

> **Note:** You can't edit an append operation.

To delete an operation:

1 In the Super Query window, select an operation.

2 On the Query menu, point to Super Queries, then click Delete Operation.

3 If an alert dialog box opens, select the branch of the join or append operation that you want to delete.

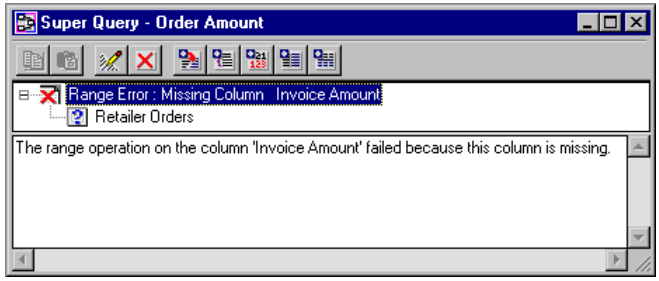
Note: To delete a component query, you must delete its join or append operation. Each branch in a join or append operation corresponds to a different component query.

Correcting Errors

Errors in the Super Query window must be corrected before the query can be saved or submitted. You can correct errors by editing either the operation or its associated component query. Most errors occur because an attribute used in an operation is missing from the component query on which the operation is performed.

To correct an error in the Super Query window:

1 In the Super Query window, click the operation causing the error. When an error occurs, BI Query displays an error icon beside the corresponding operation.


Super Query - Order Amount
Range Error : Missing Column Invoice Amount
Retailer Orders
The range operation on the column 'Invoice Amount' failed because this column is missing.

2 Read the information in the message pane. The message pane provides information about why the error occurred and how to correct it.

3 Edit the component query or operation. If you're editing more than one operation, start with the operation immediately above its corresponding query and work up through the tree.

4 Save the query.

5 To test the query, on the Query menu, click Run.

Filtering Results

For information about narrowing the range of results and performing calculations on results, see "Narrowing the Range of Results" on page 139.

You filter results in order to see different views of your data. You can specify a sort order before you submit the query or after you have results. When you specify the sort order before submitting the query, the DBMS does the sorting. You place fewer demands on the DBMS by specifying the sort order in the results, since BI Query sorts the results. However, if you specify the sort order in the results, you can't save the query as a data values query file, and you can't save the SQL for the query as a text file.

> **Note:** All forms of filtering results—sorting results after the query has been submitted, setting a range, and performing calculations on results—cause your query to become a super query when saved from a results window. You can also filter results by adding operations in the Super Query window.

Getting Results in Sorted Order

The rows in any column of results can be sorted into alphabetical, numerical, or chronological order (or the reverse), as appropriate. You can determine which columns to sort and the order in which they are to be sorted. Using the Sort dialog box, you can specify a sort order before you submit the query or after you have results.

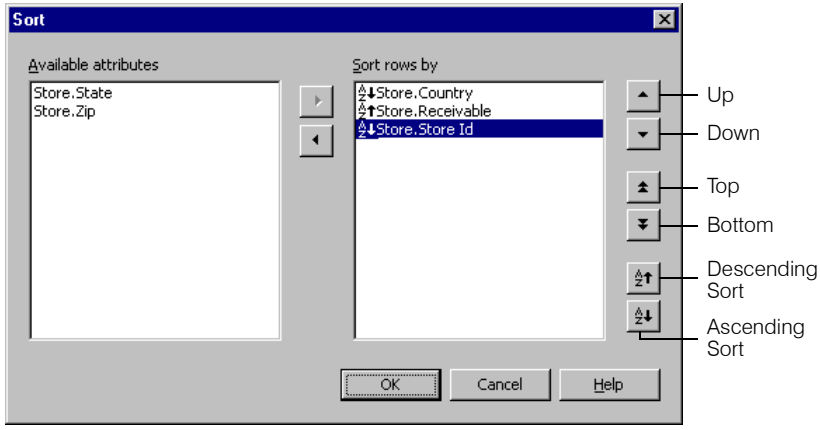
For example, if you have a column of customer names and a column of delivery dates, you can specify whether they should be organized alphabetically by customer name (with delivery dates given for each customer given chronologically) or chronologically by delivery date (with an alphabetized list of customers who received delivery for each date).

To sort results:

1 Open the Sort dialog box by doing one of the following:

- Open the Super Query window. On the Query menu, point to Add Operation, then click Sort.
- On the Results menu, point to Filter, then click Sort.
- In an attribute window, click the attributes you want to sort. On the Query menu, click Sort.

2 From the Available Columns list, click each column or attribute that you want to sort, then click the right arrow button. The selected items move to the Sort Rows By list. The top-to-bottom order of items in the Sort Rows By list corresponds to the sort order (the topmost item is sorted first).



3 To change the position of an item in the Sort Rows By list, select the item, then click the Up, Down, Top, or Bottom buttons as appropriate.

4 To sort an item in reverse alphabetical, numerical, or chronological order, select the item, then click the Descending Sort button.

5 To sort an item in alphabetical, numerical, or chronological order, select the item, then click the Ascending Sort button.

6 Click OK.

Reordering Columns

For more information on ordering columns, see "Reordering Columns Before Submitting a Query" on page 31.

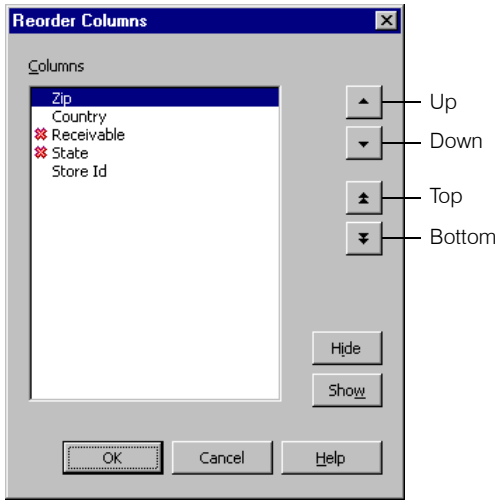
Columns appear in results windows and Standard Reports in the order in which you chose attributes while creating the query. You can change that order before submitting the query or after you have results.

To reorder columns after you have results:

1 To open the Reorder Columns dialog box, do one of the following:

- In the Super Query window, on the Query menu, select Add Operation, then click Reorder.
- In the Results window, on the Results menu, click Reorder Columns.
- With a Standard Report active, on the Report menu, click Reorder Columns.

In the Reorder Columns dialog box, the top-to-bottom order of columns in the Columns list corresponds to the left-to-right order of columns in the results window.



2 From the Columns list, select the column whose position in the list you want to change. Use the Up, Down, Top, or Bottom buttons to move the selected column to the desired position.

3 To hide a result column, select it from the Columns list, then click Hide. A red X icon appears beside the column in the list. Hidden columns do not appear in the results window or in any report based on that results window.

4 To display a hidden column, select it from the Columns list, then click Show.

> **Note:** You cannot hide or show columns in a standard report directly. You must hide or show the column in the results set or super query, then generate the report.

5 Repeat steps 2–4 until you have achieved the desired column order.

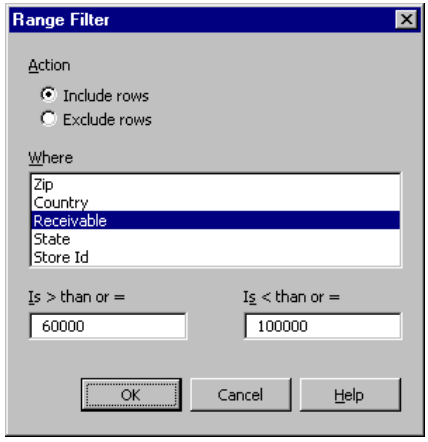
6 Click OK.

Narrowing the Range of Results

To help you analyze your data, you can focus on a particular range of results. Specifying a range for results includes only the rows within that range. For example, you can specify retailer credit limits between $200,000 and $600,000, sales received between June 30 and December 31, and salespeople's names between the letters A and S. You can also focus on results outside a range.

To narrow a range of existing results:

1 In a Results window, point to Filter on the Results menu, then click Range. The Range Filter dialog box opens.



2 From the Where list, select the column you want to filter.

3 In the Is > Than Or = box, type the minimum value for the range.

4 In the Is < Than Or = box, type the maximum value for the range.

5 Do one of the following:

- To display the results that lie within the specified range, select Include Rows.
- To display the results that lie outside the specified range, select Exclude Rows.

6 Click OK.

Note: You can also specify a range for results before submitting a query using the BETWEEN and NOT BETWEEN operators in the qualification tree. For more information, see "Finding a Range of Values" on page 48.

Combining Results Sets

For information on troubleshooting combined results, see the Help for your BI Query application.

In order to enlarge the scope of your data, you can combine results from multiple queries into one results set. You can do the following operations:

- append the rows of one results set to another
- join one or more columns of one results set to another

The results sets can be from queries submitted in different design windows within the model. Because different windows can connect to different databases, you can combine information from two different databases.

You can combine the results in two ways:

- Before you run the corresponding queries. In this case, you use the Super Query window to combine the results of the two queries.
- After you run the queries. In this case, you specify the results sets you want to combine directly.

For more information on buttons, see the *BI Query Data Models User's Guide*.

If you save the query after combining results, you can open and edit it in the Super Query window. (However, you cannot save the corresponding SQL string as a text file.) You can also link the combined query to a button to perform the combine operations automatically.

About Appended Columns

When you have similar sets of results (for instance, identical kinds of data from two different years) that you want to present together, you can append the rows of one results set to another. If the two sets have different column names, BI Query copies the name(s) from the first set to the combined set.

Rules for Appending Rows

Adhere to the following rules when you append the rows of one results sets to the rows of another:

- The data types of the columns in the first results set must match the data types of the corresponding columns in the second results set. For example, if the first results set contains two columns, and their data types are character and numeric, respectively, the first two columns of the second results set must also be character and numeric.
- The order of columns in one results set must match the order of columns in the other. One set may have additional columns that the other does not have as long as these extra columns are at the end (on the right).

Example

In the following example, the rows from Query Results 2 have been appended to Query Results 1 to form a new results set.

Query Results 1

Name	Sales ($)
Abicon Inc.	12,000
Textal Ltd.	113,000
Unisystems	10,145
Ziptech	6,430

Query Results 2

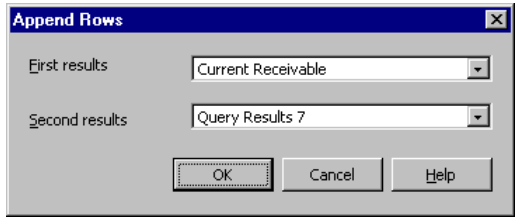
Names	Sales ($)	State
Bainstream Inc.	10,000	New York
Knatl Corp.	2,000	California
MonoLitho	2,100	Ohio
Textronics	60,000	Illinois

Appended Results

Name	Sales ($)	State
Abicon Inc.	12,000	
Textal Ltd.	113,000	
Unisystems	10,145	
Ziptech	6,430	
Bainstream Inc.	10,000	New York
Knatl Corp.	2,000	California
MonoLitho	2,100	Ohio
Textronics	60,000	Illinois

Combining Results by Appending Rows

Use the Append Rows dialog box to append one results set to another.



To append rows using existing results sets:

1. Open the results sets you want to combine, either by running the corresponding queries or by opening saved results set files.
2. On the Results menu, point to Combine, then click Append Rows. The Append Rows dialog box opens.
3. From the First Results drop-down list, select the first results set.
4. From the Second Results drop-down list, select the results set you want to append to the first set.
5. Click OK.

To append rows using queries:

1 Specify the first query by doing one of the following:

- Open an existing super query.
- Run a query. With the results set active, click Show on the Query menu.

The Super Query window opens.

2 On the Query menu, point to Add Operation, then click Append. The Append Rows dialog box opens.

3 From the Query for the Bottom Branch drop-down list, select the query whose results you want to append to the results of the first query.

4 Click OK. In the Super Query window, an append operation appears in the tree list above the component queries.

About Column Joins

For more information on creating joins between data objects, see the *BI Query Data Models User's Guide*.

You can join columns from different results sets to produce a new results set. The new results set can contain data from more than one data object (including actual relationships and query data objects) from more than one design window. Typically, the BI Query administrator specifies the join operations between data objects when the model is created. However, you can produce the same results by joining columns from different results sets. In effect, you are creating a relationship between results sets that is equivalent to a relationship between joined data objects.

By convention, a column join has a "left" results set and a "right" results set. You specify which column in the left set joins to which column in the right. (You can join more than one pair of columns.) The combined results set represents each pair of joined columns in a single column.

Reasons for Joining Columns

You may want to join columns of results sets for the following reasons:

- the design window doesn't provide joins for the attributes you need to join
- the data model doesn't include permission to create relationships between tables

- the database doesn't support outer joins
- you need to combine results from more than one database
- after examining recent results sets, you want to join certain columns but do not want to resubmit the component queries (for example, they may require too much processing time)

Rules for Joining Columns

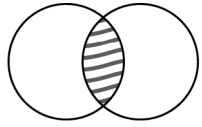
Adhere to the following rules and guidelines when joining a column of one results sets with a column of another:

- The joined columns must have the same data type. However, you can join columns containing floating point and money data types.
- If the joins you need already exist in the data model, you may want to create a query that retrieves the results rather than a super query that combines results sets. The DBMS—which can handle a single query at a time—may be able to optimize the join operations more efficiently than BI Query.

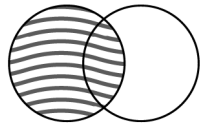
Types of Column Joins

For more information on types of joins, see the *BI Query Data Models User's Guide*.

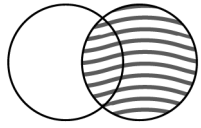
You can specify the relationship between results sets using one of four types of join: an inner join, a full outer join, a left outer join, or a right outer join:



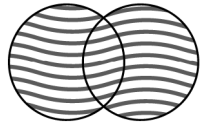
Inner join



Left outer join



Right outer join



Full outer join

Inner Join An inner join includes only those rows that have an exact match in the joined columns. (BI Query actually performs a type of inner join called a natural join, which displays the joined columns only once in the combined results set.)

Full Outer Join A full outer join includes all rows from both results sets, even if they do not match in the joined columns.

Left Outer Join A left outer join includes all rows from the left results set and those from the right that match in the joined columns.

Right Outer Join A right outer join includes all rows from the right results set and those from the left that match in the joined columns.

> **Note:** Unmatched data in a full, left, or right outer join is represented by a `NULL` in the joined column.

Example

In the following example, the first set of query results shows the total sales for products that a company has produced in the U.S. The second set of results shows the total payments received to date for all products sold by the company, including those produced outside the U.S.

An inner join of both results sets on Product ID shows all American products by the company that have been sold and received some payment to date. A left outer join on the same column shows the sales and payments of all American products, regardless of payment.

Query Results 1

Product ID	Sales ($)
101	15,000
102	52,303
213	995
331	2,561
511	8,000
514	2,303
818	612

Query Results 2

Product ID	Amount Received ($)
101	13,050
102	52,303
331	1,061
382	10,090
449	900
818	612

Equijoin on Product ID

Product ID	Sales ($)	Amount Received ($)
101	15,000	13,050
102	52,303	52,303
331	2,561	1,061
818	612	612

Query Results 1 Left Outer Joined to Query Results 2 on Product ID

Product ID	Sales ($)	Amount Received ($)
101	15,000	13,050
102	52,303	52,303
213	995	NULL
331	2,561	1,061
511	8,000	NULL
514	2,303	NULL
818	612	612

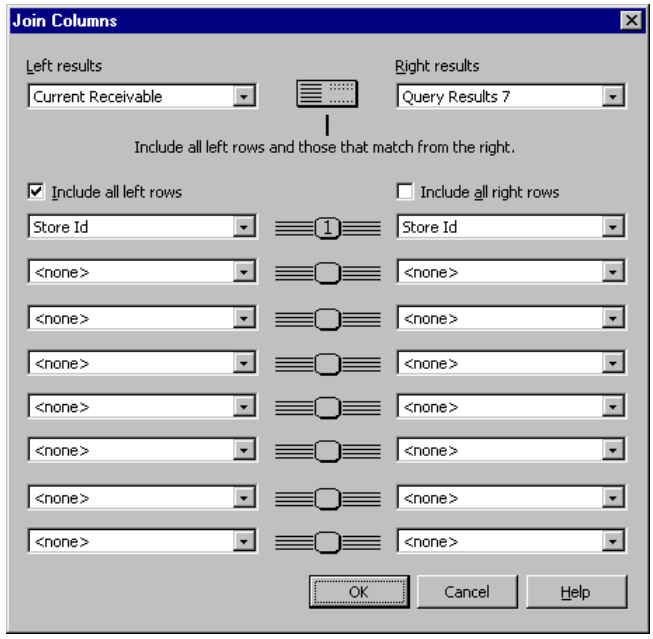
Combining Results by Joining Columns

Use the Join Columns dialog box to specify which results sets to join on which columns. You must also specify the type of column join. The joined columns must have the same (or similar) data type. You can join one or more pairs of columns. (Typically, you should join columns that represent primary keys.)

To join results-set columns:

1 Open the Join Columns dialog box by doing one of the following:

- Run one of the queries whose results you want to join. (This query will produce the "left" results set.) With the results set active, click Show on the Query menu. The Super Query window opens. On the Query menu, point to Add Operation, then click Join.
- Open the two results sets you want to join. On the Results menu, click Combine, then click Join Columns.



2 Do one of the following:

- If you are joining results sets directly, select the "left" results set from the Left Results drop-down list, then select the "right" results set from the Right Results drop-down list.
- If you are using the Super Query window to join results, select the "right" query from the Query For The Bottom Branch drop-down list. (The "left" query is the super query.)

3 Do one of the following:

- To create an inner join, clear the Include All Left Rows and Include All Right Rows check boxes. (This is the default setting.)
- To create a left outer join, select Include All Left Rows.
- To create a right outer join, select Include All Right Rows.
- To create a full outer join, select both check boxes.

4 Select the columns you want to join (up to eight pairs) in the order in which you want to join them.

> **Note:** A joined column appears only once in the combined results set.

5 Click OK.

Chapter 9

Creating Standard Reports

About Standard Reports **151**
Standard Report Structure 152
Data Formats in Standard Reports 153

Basic Report Tasks **153**
Creating Standard Reports 153
Saving Report Specifications 154
Applying Report Specifications 156
Printing and Exporting Standard Reports 157

Adding Calculations to Standard Reports **157**
Adding Subtotals 158
Adding Grand Totals 159
Adding Calculated Columns 160

Manipulating Columns in Standard Reports **162**
Resizing Report Columns 162
Combining Columns 163
Banding Columns 164
Stacking Columns 165
Formatting Column Text and Headings 166
Customizing Column Settings 167

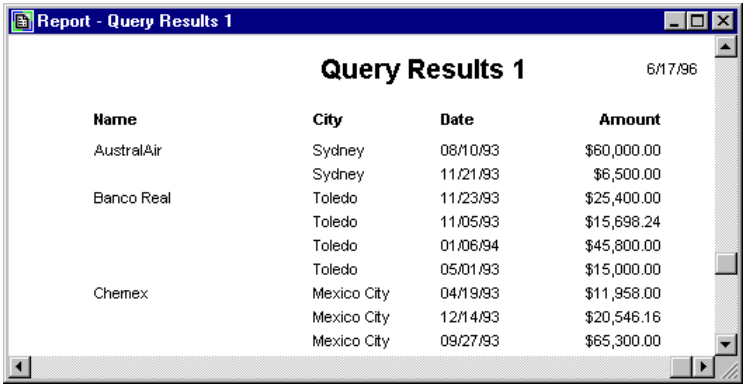
Adding Borders to Standard Reports **168**

Using Ornaments in Standard Reports **169**
Special Text Strings in Standard Reports 170
Adding Text Ornaments to Standard Reports 171
Editing Default Ornaments in Standard Reports 172

Converting Standard Reports **173**
Converting Single Standard Reports 174
Converting Multiple Standard Reports 174
Setting Batch Converter Options 175

About Standard Reports

BI Query includes a fully integrated reporting facility called Standard Reports. Using Standard Reports, you can produce fully formatted, professional reports from the results of your queries. Each standard report you create appears in a report window in BI Query, where you can format it in a variety of ways. The results in a standard report correspond to the results in a results window.



A standard report window (default format)

A standard report consists of two components:

- a query that generates the results that appear in the report
- a report specification that specifies how the results are formatted and what additional calculations are applied to them

If the BI Query administrator has provided report specifications with the data model, you can use them to create a report from query results. You can also create reports using your own specifications or the default specification.

For more information, see "Publishing Standard Report Specifications" on page 241.

If you are working in a BI Server environment, you can publish your standard reports to the BI Server Repository. BI Web users can then access and run the reports over the Web. Other BI Query users can also access the reports when they retrieve the data model from the Repository.

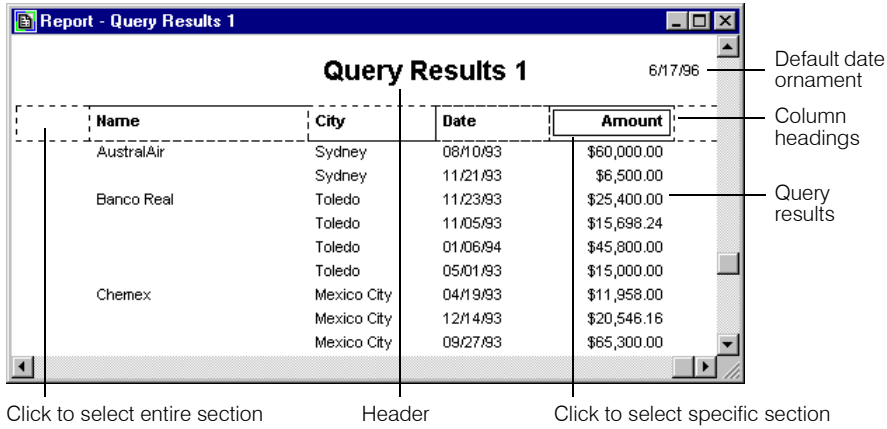
Note: Standard Reports is an older and less sophisticated reporting facility than BI Query Reports. However, you can convert standard reports into BI Query Reports format. For more information, see "Converting Standard Reports" on page 173.

Standard Report Structure

For more information on using ornaments, see the *BI Query Data Models User's Guide*.

Standard reports are divided into the following sections:

Section	Description
Header	Contains the title and date and appears at the top of each page of the report. You can enlarge it, change its appearance and content, and add ornaments to it.
Column headings	Correspond to the headings in the results set. You can change their appearance, content, width, and depth. You can also add ornaments to them.
Results	Contain the actual results of the query. You can add subtotals, grand totals, and ornaments to them. You can reorder them, replace their values, and change their widths, appearance, and data formats.
Footer	Contains the page number and appears at the bottom of each page of the report. You can enlarge it, change its appearance and content, and add ornaments to it.



When you click in a section, a border appears around it. When a section is too small to accommodate the information in it or when it displays too much empty space, you can resize it to improve the layout and ensure that all the information appears when you print the report.

Data Formats in Standard Reports

For more information on formatting standard reports, see "Formatting Data: The Basics" on page 60.

When BI Query returns results from the database, the format in which they appear may not be the way you want them represented in a standard report. You can change the default display format for each attribute in a standard report.

If you save a standard report specification with a query and subsequently change the display format for an attribute in the query, the original display format in the corresponding standard report does not change. To apply the new format to the report, change the format for the corresponding attribute in the report window, then save the report specification with the query.

Basic Report Tasks

This section describes the basic tasks you need to perform to create and save a standard report. Later sections describe how to add calculations to a standard report, how to manipulate its columns, and how to change its appearance.

Creating Standard Reports

When you create a standard report, the report window displays the results of a query in a default format. You can change the appearance of a report in a variety of ways and add information to it. The title of the report corresponds to the query that generated the results.

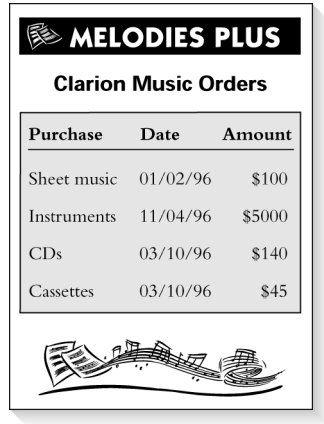
To create a standard report:

1 Retrieve, open, or generate the results you want to present in a standard report.

2 On the Results menu, point to Show as Report, then click BI Query Standard.

Saving Report Specifications

Report specifications help you present information consistently and adhere to corporate standards (such as corporate logos and fonts). If you create similar reports on a regular basis, you can use a report specification to avoid formatting the data each time.



If you create a button that links to the query and sends output to a results window, BI Query runs the query whenever you click the button, then creates the standard report using the associated specification.

When you save a report specification locally, BI Query automatically adds the extension `.rpt` to the file name and stores the specification file in the `Queries` folder (unless you export the specification). By default, BI Query also saves the corresponding query and stores it in the `Queries` folder with the same name but with the extension `.qry`. When you submit the query, BI Query automatically displays the report using the stored report specification.

If you save a query and a report specification separately and use the same name (except for the extension), BI Query automatically applies the specification to the results each time you submit the query.

> **Note:** If BI Web users run a query that has a specification associated with it, BI Web presents only the query results. To view the results as a standard report, BI Web users must open the report directly.

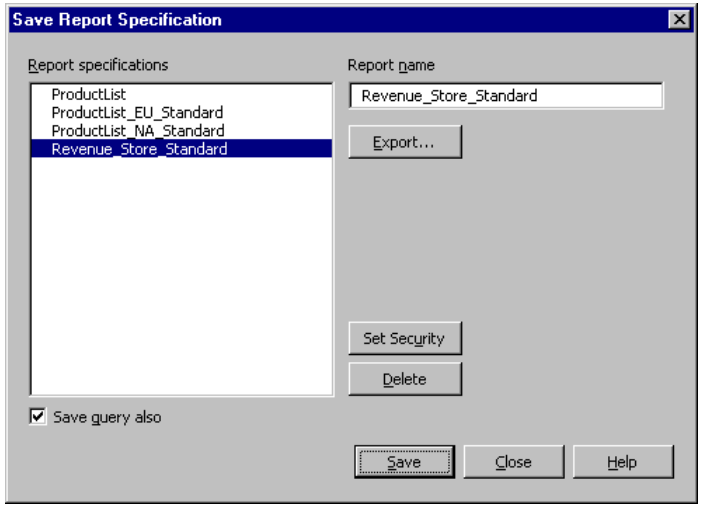
You can also remove the specification associated with a query to avoid generating a report each time you run the query. (With the query open, click Remove Report Specification on the Query menu.)

You can save queries and report specifications only if the Save Queries permission is assigned.

To save a report specification:

1 Create and format the report. Add any required calculations to it.

2 On the Report menu, click Save Report Specification. The Save Report Specification dialog box opens.



3 In the Report Name box, type the file name for the specification. Do not include the .rpt extension (BI Query adds it automatically).

4 If you don't want to save the corresponding query, clear Save Query Also.

5 Do one of the following:

- To save the specification to the Queries folder, click Save. BI Query also saves the corresponding query to the same location if Save Query Also is selected.

- To export the report specification to a location other than the Queries folder, click Export. In the Export Report Specification dialog box, browse to the location where you want to save the file, then click Save.

You can also publish standard reports to the BI Server Repository. Published reports can be opened by BI Web users. For more information, see "Publishing Standard Report Specifications" on page 454.

Applying Report Specifications

After you've saved a report specification, you can apply it to the results of other queries. When you select a specification, BI Query compares the columns in it with the new results. If the corresponding tables and attributes match, BI Query formats the columns accordingly. If they don't match, BI Query formats the columns based on their position in the report. (For example, it applies the format of the first column in the report specification to the first column of results.)

To apply a report specification to results:

1 Retrieve, open, or generate the results you want to present in a standard report.

2 On the Results menu, point to Show as Report, then do one of the following:

- To use a specification that is stored on your local machine or network, click Open Report Specification. The Open Report Specification dialog box opens.
- To use a specification stored in the BI Server Repository, click Retrieve Report Specification. (You must be in a BI Server environment to use this menu item.) The Retrieve Report Specification dialog box opens.

3 In the dialog box, double-click the specification you want to apply to the current results set.

Printing and Exporting Standard Reports

When you print a report, you can include a title page that provides the date, time, number of pages, and list of prompt values used in the query. Before printing the report, click Title Page on the Report menu.

You can print a standard report to a printer or a file, or export it directly to a text file. Standard reports print exactly as they appear on screen.

To print a report:

1 Create the report or run the query associated with it.

2 On the File menu, click Print. The Print dialog box opens.

3 Do one of the following:

- To print to a printer, specify the options you want, then click OK.
- To print to a file, select Print to File, then click OK. In the Print To File dialog box, specify a name for the file, then click OK.

To export a report to a text file:

1 Create the report or run the query associated with it.

2 On the Report menu, click Export Report.

3 In the Export Report to File dialog box, specify a location and name for the file.

4 Click Save.

Adding Calculations to Standard Reports

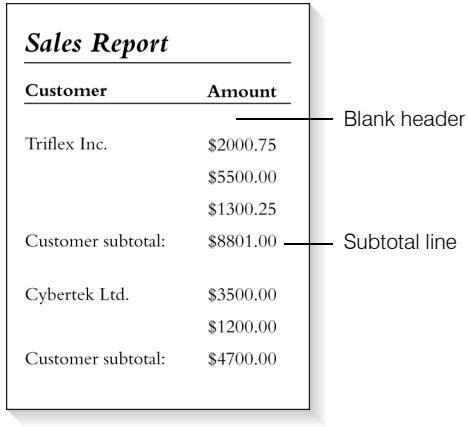
For more information on calculated attributes, see "Creating Calculated Attributes" on page 186.

You can add subtotals and grand totals to standard reports to perform standard calculations (sum, average, and so on) or custom calculations (such as squaring the sum of a column). You can also add calculated columns—similar to calculated attributes—that perform user-defined calculations.

Unlike calculated columns, calculations on totals do not appear in a separate column; instead, they appear on separate subtotal or grand total lines.

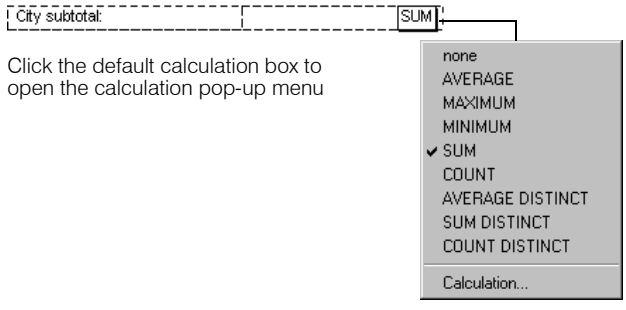
Adding Subtotals

Subtotals allow you to analyze data after results are returned. A subtotal line appears below each group of information in the column, displaying a label and default calculation. The label is an ornament; you can change its content and appearance (font, size, and so on). You can also move the label to a different location or delete it. When you add a subtotal, a blank header is added above each group of information. You can add ornaments to the header and resize it.



To add subtotals:

1. Click the column on which to base the subtotal (usually a column that's been sorted).
2. On the Report menu, click Add Subtotal.
3. Click the default calculation box in one of the subtotals.



4 From the pop-up menu that opens, do one of the following:

- To apply a standard calculation, click it.
- To specify your own calculation, click Calculation, then create the calculation in the Edit Subtotal Calculation dialog box.

For more information on calculated columns, see "Adding Calculated Columns" on page 160.

5 Click outside the subtotal to run the calculation.

> **Note:** The Count operator counts only the visible values in a column. That is, if you've suppressed duplicates, the count doesn't include the suppressed values.

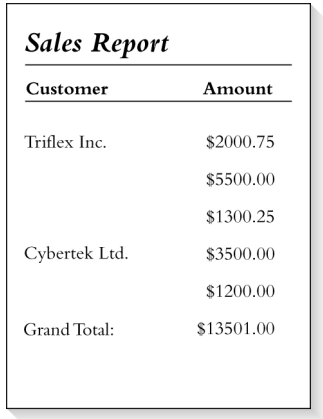
To insert a page break after each subtotal in a report:

1 Click in a subtotal line.

2 On the Report menu, click New Page After Subtotal.

Adding Grand Totals

You can add one or more grand totals to a report to help you analyze data after results are returned. For example, you may wish to calculate both the sum and the average for a numeric column.



Sales Report

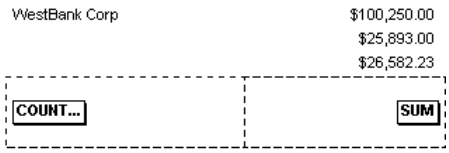
Customer	Amount
Triflex Inc.	$2000.75
	$5500.00
	$1300.25
Cybertek Ltd.	$3500.00
	$1200.00
Grand Total:	$13501.00

To add grand totals:

1 On the Report menu, click Add Grand Total.

2 For each column, do the following:

a) Click the default calculation at the bottom of the report.



For more information on calculated columns, see "Adding Calculated Columns" on page 160.

b) From the pop-up menu that opens, click a standard calculation to apply to the column. Alternatively, to specify your own calculation, click Calculation, then create the calculation in the Edit Grand Total Calculation dialog box.

3 Click outside the grand total to run the calculation.

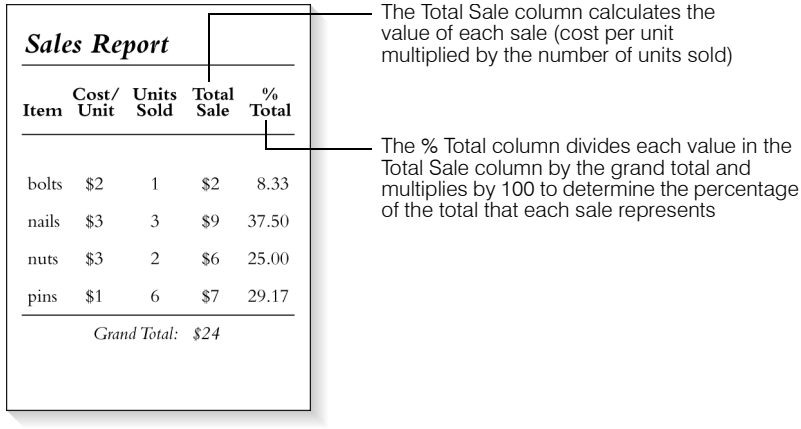
Adding Calculated Columns

You can add a column to a report to perform a specific calculation based on information in the report (such as values in a numeric column, subtotals, and grand totals).

For example, suppose your report contains a list of countries, order amounts for each country, and a grand total of those orders. You can create a calculated column to determine what percentage of the grand total each order amount represents.

You can also use multi-pass reporting to create a column that performs a calculation on other calculated columns. In this case, BI Query must make two or more passes through the data to arrive at the final calculation.

For example, suppose your report contains a list of products, the cost per unit of each product, the number of units sold, a calculated column that determines the value of each sale (the cost per unit multiplied by the number of units sold), and a grand total of sales. You can add another calculated column to determine what percentage of the grand total each sale represents.

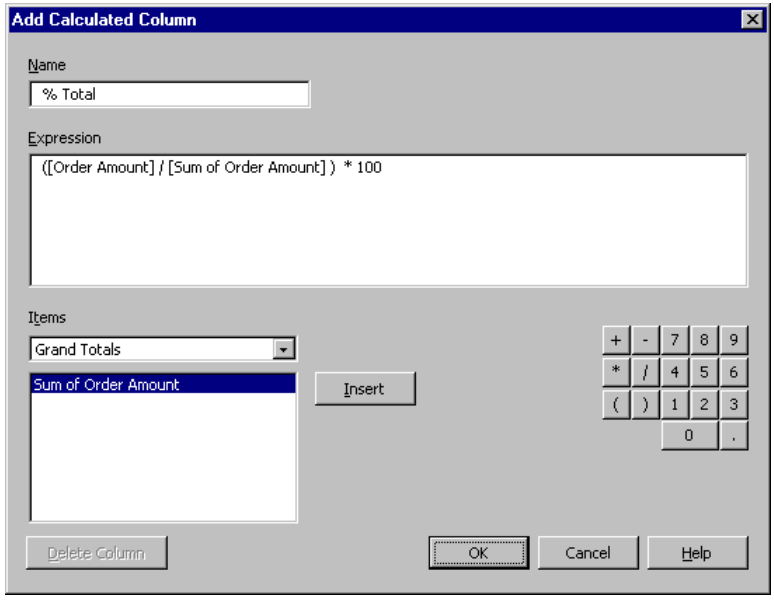

Sales Report
Item
Cost/ Unit
Units Sold
Total Sale
% Total
bolts $2 1 $2 8.33
nails $3 3 $9 37.50
nuts $3 2 $6 25.00
pins $1 6 $7 29.17
Grand Total: $24
The Total Sale column calculates the value of each sale (cost per unit multiplied by the number of units sold)
The % Total column divides each value in the Total Sale column by the grand total and multiplies by 100 to determine the percentage of the total that each sale represents

For information on editing or deleting a calculated column, see the Help for your BI Query application.

To add a calculated column:

1 On the Report menu, click Add Calculated Column. The Add Calculated Column dialog box opens.

2 In the Name box, type the name of the new column.



3 In the Expression box, type the calculation for the new column. You can also do the following:

- Use the numeric and operator keypad at the right of the dialog box to enter arithmetical expressions.
- Use the Items drop-down list box to select an item on which to base the calculation (Columns, Subtotals, Grand Totals, Prompts). From the list below the Items drop-down list box, select the calculation, attribute, or prompt that you want to add to the calculation. Click Insert.

Your calculation can involve more than one item in the report or corresponding query.

4 Click OK.

Manipulating Columns in Standard Reports

When you generate a standard report from the results of a query, BI Query displays the results in the order in which they appear in the results window. If you want to reorganize the results or save space, you can change the column order and combine, band, or stack columns.

Resizing Report Columns

When data appears as a sequence of asterisks, the corresponding column is not wide enough to display the data.

If a standard report contains more columns than fit on one page, BI Query spreads the columns over the required number of pages; a gray-shaded line separates each page. To view each column, scroll to the right. To prevent columns dividing over more than one page, resize or reorder them as necessary.

You can also reorder columns. For more information, see "Reordering Columns" on page 137.

To resize columns:

1 Click a column.

2 Do one of the following:

- Drag the right-hand border of the column to the width you want.
- On the Report menu, click Column Settings. The Column Settings dialog box opens. From the Columns list, select the column you want to resize, then modify its character width in the Column Width box. Click OK.

Combining Columns

When related data (such as a person's name and title) are in different columns and you want to present them together, you can combine the columns. You can combine two or more columns into a single column, and you can merge combined columns with another column. When BI Query combines columns, the new column displays the heading of the first selected column.

Sales Report

Surname	Title
Czerny	VP Sales
Plainsfield	VP Marketing
Hamford	District Mgr.
Jorgenson	Regional Mgr.
Lifkin	Sales Rep.
Machu	Regional Mgr.
Bruining	Distribution Mgr.
Millard	Account Dev.

Original

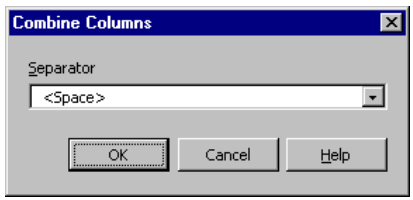
Sales Report

Surname
Czerny, VP Sales
Plainsfield,VP Marketing
Hamford, District Mgr.
Jorgenson, Regional Mgr.
Lifkin, Sales Rep.
Machu, Regional Mgr.
Bruining, Distribution Mgr.
Millard, Account Dev.

Combined

To combine columns:

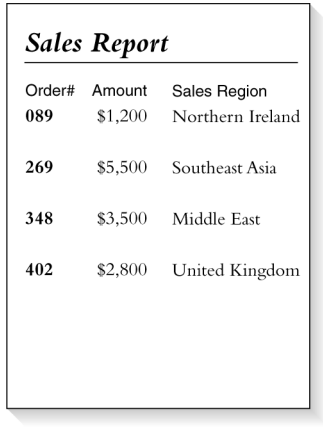
1 Shift + click the columns you want to combine. On the Report menu, click Combine Columns. The Combine Columns dialog box opens.



2 From the Separator drop-down list box, select the separator you want to appear between the items in each combined column (for example, a space). You can also type a separator in the box. Click OK.

Banding Columns

To create a standard report in which one column occupies the entire width of the page, you can band the column. This feature is useful if a report contains several columns with short pieces of data and one column that contains a lengthy description. You can band the descriptive column so that it appears across the page, either above or below the other columns. Banding columns also lets you create subheadings in a report.



Sales Report

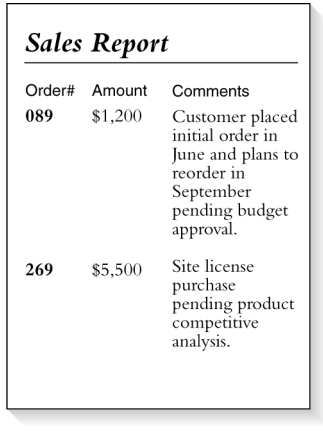
Order#	Amount	Sales Region
089	$1,200	Northern Ireland
269	$5,500	Southeast Asia
348	$3,500	Middle East
402	$2,800	United Kingdom

Original

Sales Report

Order#	Amount
Northern Ireland	
089	$1,200
Southeast Asia	
269	$5,500
Middle East	
348	$3,500
United Kingdom	
402	$2,800

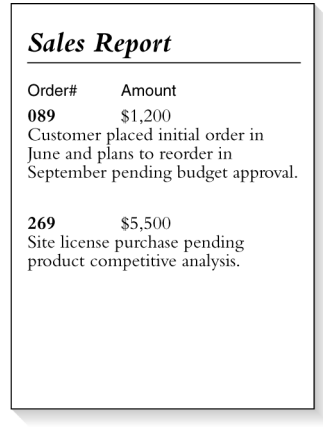
Banded (with Before selected)



Sales Report

Order#	Amount	Comments
089	$1,200	Customer placed initial order in June and plans to reorder in September pending budget approval.
269	$5,500	Site license purchase pending product competitive analysis.

Original



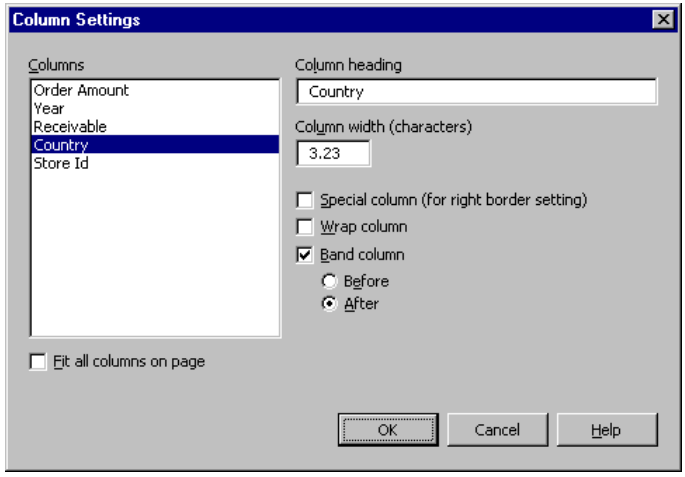
Sales Report

Order#	Amount
089	$1,200
Customer placed initial order in June and plans to reorder in September pending budget approval.	
269	$5,500
Site license purchase pending product competitive analysis.	

Banded (with After selected)

To band a column:

1 On the Report menu, click Column Settings. The Column Settings dialog box opens.



2 From the Columns list, select the column you want to band.

3 Select Band Column, then select Before or After to display the column above or below the other columns.

4 Click OK.

Stacking Columns

Instead of displaying columns horizontally across a page, you can stack them into a single vertical column, with each item appearing on a separate line; the columns that you don't stack appear in their normal position. The stacked column is as wide as the widest column in the stack. Only the heading of the left-most column in the stack is used.

A standard report can contain only one stacked column. You can add columns to a stacked column by un-stacking the column, then re-stacking. You can include both banded and stacked columns in a report. Stacked columns appear to the left of banded columns. The stacked column displays the heading of the first column you select.

Original Stacked

To stack columns:

1 Shift + click the columns you want to stack.

2 On the Report menu, click Stack Columns.

Formatting Column Text and Headings

Varying the appearance of text in a report provides variety and contrast, improves readability, and helps distinguish between different pieces of information. A unique typeface can make a title or masthead stand out. Bolding and italicizing add further contrast and emphasize important information.

You can format the text in columns and column headings in terms of font, size, style, alignment, effects (strikeout and underline), and color. You can also change the actual text of a column heading (which is initially the same as the column heading in the corresponding results set).

To format text in a column or column heading:

1 Click the column or column heading you want to format.

2 On the Report menu, click Text Style. The Text Style dialog box opens.

3 Specify the text format for the selected column or heading.

4 If you are formatting a column heading and want to change the text of the heading, type the new heading in the Text box.

5 Click OK.

Customizing Column Settings

In addition to resizing and banding a column, you can apply the following customizations:

- modify the column name
- change the text wrap within the column
- make the column fit the page
- add a special border to the column

To customize a column:

1 On the Report menu, click Column Settings. The Column Settings dialog box opens.

2 From the Columns list, click the column you want to customize.

3 To change the title for the column, type the new title in the Column Heading box.

For more information on border formats, see "Adding Borders to Standard Reports" on page 168.

4 To apply a special border to the left side of the column, click Special Column. You can format this special border using the Report Borders for Standard Reports dialog box.

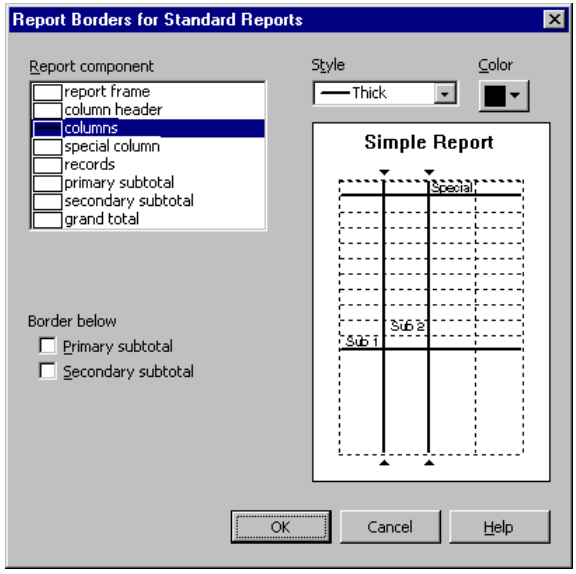
5 To wrap column data that exceeds the width of the column, select Wrap Column. This check box is enabled only for columns containing character data.

6 Click OK.

Adding Borders to Standard Reports

You can also create borders manually using the Line tool on the Drawing toolbar. For more information on report borders, see the Help for your BI Query application.

Using the Report Borders for Standard Reports dialog box, you can add horizontal and vertical borders to report components to help you organize and highlight information. For example, you can add vertical lines between columns or horizontal lines below column headings or between records. You can set a default border format for all standard reports; you can also specify borders for a particular report.



To add borders to a standard report:

1 Open the Report Borders for Standard Reports dialog box by doing one of the following:

- On the Tools menu, click Preferences. In the Preferences dialog box, click Report Borders.
- Create the report. On the Report menu, click Borders.

Note: If you open the Report Borders for Standard Reports dialog box for a particular standard report, the border format you specify applies only to that report. If you open the dialog box from the Preferences dialog box, the border format applies to all standard reports that you subsequently create.

2 From the Report Component list, select the component you want to format.

3 From the Style drop-down list, select the line style for the border.

4 To specify a color for the border, click the Color button and select the color from the palette that opens. To specify a custom color, click Custom Color in the palette.

5 Repeat steps 2–4 for all other report components that you want to format.

6 To add a border below any primary and secondary subtotals, select the Primary Subtotal and Secondary Subtotal check boxes.

7 Click OK.

Using Ornaments in Standard Reports

For more information on ornaments, see the *BI Query Data Models User's Guide*.

Some default report elements are text ornaments and appear in specific sections of a report by default. The title and date are text ornaments that appear in the header of a report. The title is based on the name of the corresponding results window. The date is the same as the date on your computer. The page number is a text ornament that appears in the footer of a report.

You can change the appearance and content of each default ornament, move it to a different position, or remove it altogether. For example, you can change the default title of your report to reflect the information it contains, and you can change the date to display a different date format.

You can also add text and graphical ornaments to a standard report. You create these ornaments in much the same way as you create ornaments in a design window in BI Query.

Special Text Strings in Standard Reports

The content of a text ornament in a standard report can contain any of the special text strings that BI Query supports. For example, the default date ornament contains the special string `&d`, which displays the current date.

BI Query supports the following lowercase strings in text ornaments in standard reports:

String	Description
`&t`	Displays the current time.
`&d`	Displays the current date.
`&d`*n*	Displays the current date using predefined format *n*, where *n* is a number greater than or equal to zero (0). *n* corresponds to a predefined format in the Date Component list in the Format Preferences dialog box. For example, `&d0` displays the current date using the first predefined date format, `&d1` displays the date using the second predefined format, and so on. `&d` is equivalent to `&d0`. *n* corresponds to a named predefined format only. The <Regional Settings>, <Custom>, and <Blank> formats do not apply in this case.
`&p`	Displays the current page number. The default page number ornament contains the text `Page &p`.
`&pp`	Displays the total number of pages. To display a page number in the form "Page 1 of 3", type `Page &p of &pp` in the text ornament.
`&q`	Displays the name of the query associated with the report. If the query hasn't been saved, the ornament displays the word "Untitled".
`&«`*promptx*`»`	Displays the values of the prompt called *promptx*. For example, `&«Country»` displays the values entered for the Country prompt. This string is useful if you need to include an ornament that identifies the information on which the report is based (for example, Shipments to: Singapore, where Singapore is the value supplied for the prompt). Prompt values also appear on the report title page.

You can change the content and appearance of these strings as you would any other text in an ornament.

> **Note:** To create a text ornament or button that contains a literal version of one of the above strings, precede the string with an extra ampersand (for example, "peas&&queues" displays as "peas&queues").

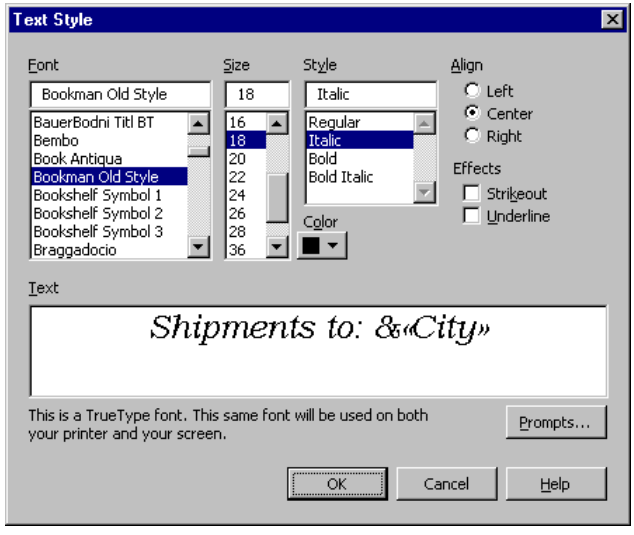
Adding Text Ornaments to Standard Reports

For more information on report sections, see "Standard Report Structure" on page 152.

When you add an ornament to a page in a multi-page report, it appears on all the pages above and below that page. If you add an ornament to a particular section of a report, such as a subtotal or the blank header above a subtotal, it's reproduced in each corresponding section.

To add a text ornament to a standard report:

1. In the report window, right-click where you want the ornament to appear.
2. On the pop-up menu that opens, click Create Ornament.
3. In the Create Ornament dialog box, click Edit Text. The Text Style dialog box opens.



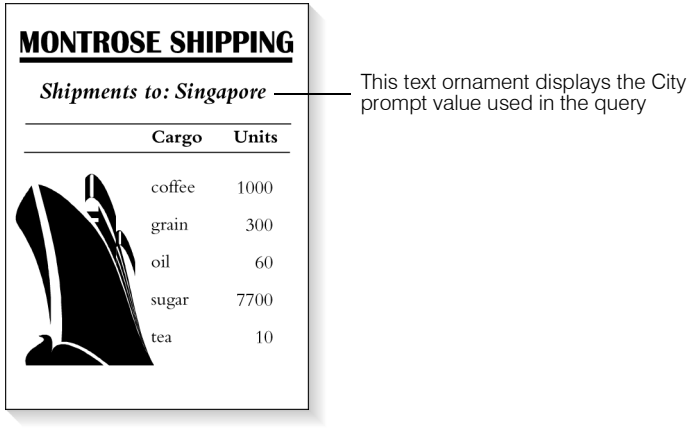
4 In the Text box, type the text for the ornament. You can include any of the special text strings supported in Standard Reports. For more information, see "Special Text Strings in Standard Reports" above.

5 To add prompt values, click Prompts. In the Prompts dialog box, double-click the name of the prompt used in the query.

The prompt appears in the Text box preceded by an ampersand (&) and delimited by chevron characters (» and «).

6 Specify the format for the ornament, then click OK.

7 In the Create Ornament dialog box, click OK.



Editing Default Ornaments in Standard Reports

Use the Edit Ornament dialog box to edit the default title, date, and page number ornaments.

To edit a default ornament in a standard report:

1 In the report window, double-click on the ornament.

2 In the Edit Ornament dialog box, click Edit Text. The Text Style dialog box opens.

3 In the Text box, edit the text for the ornament as necessary. You can include any of the special text strings supported in Standard Reports.

4 Specify the format for the ornament, then click OK.

5 In the Edit Ornament dialog box, click OK.

Converting Standard Reports

You can convert standard reports to BI Query Reports format using the Report Converter utility. Once you have converted a standard report, you can open and modify it in BI Query Reports.

Before you can use the Report Converter, you must install it. (BI Query Reports must also be installed.)

> **Note:** BI Query Reports treats stacked columns as separate items. For example, if you move one column, the other columns don't move with it. When the converter creates stacked columns, it increases the size of the detail band to accommodate the columns in the stack, then aligns the columns vertically.

To install the Report Converter:

1 Insert the Hummingbird BI or BI Query CD into your CD-ROM drive.

2 Using Windows Explorer, browse to the following folder on the CD-ROM:

```
Utilities\BI-Query\Report Converter
```

3 Double-click setup.exe.

If the error message "The referenced file or dll is not a valid Windows NT image" appears while you're installing the converter on Windows NT, you must install the converter using the /32 switch. Do the following:

1 On the Windows Start menu, click Run. The Run dialog box opens.

2 In the Open box, type the path and file name of the setup.exe file in the Utilities\BI-Query\Report Converter folder. Alternatively, click Browse and use the Browse dialog box to locate and select the file.

3 In the Open box, at the end of the path and file name of the setup.exe file, type a space followed by /32.

4 Click OK.

Converting Single Standard Reports

Once you have installed the Report Converter, you can convert a standard report to BI Query Reports format. The new report automatically opens in BI Query Reports.

To convert one report at a time:

1 In BI Query, open and submit the query associated with the standard report. Close the report window.

2 On the Results menu, point to Show as Report, then click BI/Query Standard Reports Converter. The Open dialog box opens.

3 In the dialog box, locate and select the report specification you want to convert, then click Open. (By default, BI Query saves standard report specifications in the Queries folder. Specifications have the extension .rpt.)

4 In BI Query Reports, check the newly converted report and make any necessary changes. The converted report has the same name as the specification, except that it has an .rep extension. The converter saves the converted report in the same folder as the specification.

Converting Multiple Standard Reports

You can convert multiple reports simultaneously using the batch converter utility.

To convert multiple reports using the batch converter:

1 Using Windows Explorer, browse to the folder where BI Query has been installed. By default, this folder is the following:

```
C:\Program Files\Hummingbird\BI
```

2 In the folder, locate and open the ReportsConverter subfolder, then double-click Rpt2Rep.exe. The BI/Query Standard Reports Converter dialog box opens.

3 On the Files page, click Model.

4 In the Select BI/Query Data Model dialog box, locate and select a data model, then click Open.

5 Do one of the following:

- To add reports to the list of files to be converted, click Add. In the Select Files to Convert dialog box, locate and select the reports, then click Open. You can use the Shift and Control keys to select more than one file, but you can add only the reports that are associated with the selected data model.
- To add all of the standard reports associated with a data model, click Add All. In the Add All Reports From Directory dialog box, specify the directory containing the date model, then select Include Subdirectories. (Alternatively, in the Directory box, type the explicit path for the folder containing the report specifications.) Click OK.

6 Click Convert. If the query for a standard report contains a prompt, the prompt appears while the report is being converted. In this case, specify the prompt values you want.

Setting Batch Converter Options

The batch converter keeps a log file of the conversion process. The log file lists the changes made during the conversion and keeps a record of any problems that occurred.

To view or open the log file:

1 In the BI/Query Standard Reports Converter dialog box, click the Log tab. The log appears in the Log page.

2 To open the log file, click Open.

By default, the batch converter saves the converted files in the same folder as the original standard report specifications. You can specify alternative destinations for the temporary conversion files and the converted files.

To specify a destination folder for the new BI Query Reports files:

1 In the BI/Query Standard Reports Converter dialog box, click the Options tab.

2 In the Default Destination Directory box, type the full path of the destination folder you want.

3 Click Close.

4 Restart the converter. (You must restart the converter for your changes to take effect.)

5 Convert the files. For more information, see ""Converting Multiple Standard Reports" on page 174".

To change the working directory for the temporary conversion files:

1 In the BI/Query Standard Reports Converter dialog box, click the Options tab.

2 In the Working Directory box, type the full path of the folder you want. Make sure that the working folder is not the same as the destination folder.

3 Convert the files.

Chapter 10

Building Advanced Queries

Applying Functions to Attributes **179**
Built-In Aggregate Functions 180
Types of Grouping 181
Applying a Function After Grouping 182
Applying a Function Before Grouping 184
Grouping without Functions 185

Creating Calculated Attributes **186**
Using Functions in Query Expressions 189

Using Subqueries **190**
Creating and Inserting Subqueries 191

Using Data Sets **192**
Data Set File Syntax 193
Using a Data Set to Qualify an Attribute 194
Viewing Data Sets 194
Using a Data Values Results File as a Data Set 195

Creating Tables in the Database from Query Results **196**

Creating Dynamic Relationships **197**
Deleting a User-Created Relationship 201

Using Outer Joins in a Query **201**
Specifying Join Order 203
About Join Conflicts 204
About Outer-Join Syntax in Queries 206
Changing Outer-Join Syntax for a Connection 206

Modifying the SQL for a Query **207**
Using SQL Prefixes and Suffixes 209

Building Circular Queries **210**
Circular Queries with a Correlation 210
Building a Circular Query with a Correlation 213
Circular Queries without a Correlation 214
Building a Circular Query without a Correlation 215

Applying Functions to Attributes

You can apply a function to an attribute to perform a basic calculation such as summing or averaging results. You can also apply a user-defined calculation to an attribute to carry out more sophisticated calculations.

Aggregate functions are mathematical operations that allow you to calculate summary values from a set of values for an attribute. For example, you can retrieve retailers' total sales by applying the SUM function to an Invoice Amount attribute. Different functions are available depending on the attribute, its data type, and the DBMS you're using.

When you apply an aggregate function to an attribute, BI Query automatically groups all the other attributes in the query in the order in which you selected them. (If this order isn't appropriate to your needs, you can change it.)

Grouping organizes the data into sets and retrieves a summary value for each set. All selected attributes to which functions haven't been applied are grouped to ensure that the results set doesn't calculate a summary value for individual members of the group.

Example

A database contains the following data about retailer sales:

Name	City	Invoice Amount
Sterling	Ottawa	$3000
Triteck	Toronto	$1000
Billings	Toronto	$3000
Sterling	Ottawa	$1000
Billings	Toronto	$4000
Saddler	Toronto	$5000
Triteck	Ottawa	$2000
Triteck	Toronto	$3000

If you want to retrieve a list of retailers, cities, and total retailer sales, you could form a query in which you select the Name, City, and Invoice Amount attributes (in that order) and apply the SUM function to Invoice Amount.

When you submit the query, the DBMS organizes the data into groups (first by Name, then by City). Each row of results in a group has identical values except the attribute to which the function is applied.

The table below shows how the DBMS groups the data. The rows for "Billings, Toronto" represent one group, the row for "Saddler, Toronto" represents another group, and so on.

Name	City	Invoice Amount
Billings	Toronto	$3000
Billings	Toronto	$4000
Saddler	Toronto	$5000
Sterling	Ottawa	$1000
Sterling	Ottawa	$3000
Triteck	Ottawa	$2000
Triteck	Toronto	$1000
Triteck	Toronto	$3000

The DBMS then applies the SUM function to each group. Most DBMSs sort each group before returning the results. In this case, groups are sorted first alphabetically by name, then alphabetically by city.

Name	City	Invoice Amount
Billings	Toronto	$7000
Saddler	Toronto	$5000
Sterling	Ottawa	$4000
Triteck	Ottawa	$2000
Triteck	Toronto	$4000

Built-In Aggregate Functions

BI Query provides the following built-in functions:

Function Name	Operation
AVERAGE	The average of all the values of an attribute or of each group.

Function Name	Operation
MAXIMUM	The maximum value of an attribute or of each group. The maximum is the last alphabetical item, the most recent date, or the highest number.
MINIMUM	The minimum value of an attribute or of each group. The minimum is the first alphabetical item, the earliest date, or the lowest number.
SUM	The sum of all the values of an attribute or of each group.
COUNT	The number of values that are not NULL (missing) values of an attribute or group.
AVERAGE DISTINCT	The average of all the values of an attribute or of each group, ignoring any duplicates.
SUM DISTINCT	The sum of all the values of an attribute or of each group, ignoring any duplicates.
COUNT DISTINCT	The number of values that are not NULL (missing) values of an attribute or group, ignoring any duplicates.

Types of Grouping

Grouping can be done either before or after the calculation, depending on the results you want.

Grouping Before Calculating When you group results before applying a function, BI Query eliminates the records that don't meet the specified qualification, groups the results, then performs the calculation.

For more information on calculated attributes, see "Creating Calculated Attributes" on page 186.

Grouping After Calculating When you group results after applying a function to an attribute, you qualify the attribute with a group qualification. The qualification is applied to the entire query, after all aggregate functions have been calculated (including any calculated attributes that are treated as aggregates) and the appropriate attributes have been grouped. BI Query allows you to apply a group qualification to query results obtained using aggregate functions such as SUM, AVERAGE, and COUNT.

When you group results before calculating, the SQL for the query contains a `WHERE` clause. When you group results after calculating, the SQL for the query contains a `HAVING` clause.

Examples

In order to retrieve retailers' largest sales, you might include the Retailer Name and Invoice Amount attributes in the query, qualify Invoice Amount to be greater than $50,000, then apply the SUM function to it. In this case, BI Query finds all sales that are greater than $50,000, groups them by Retailer Name, then sums them.

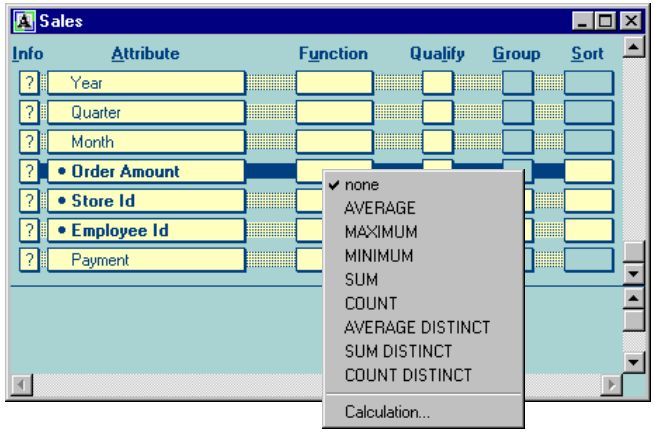
You could also apply the SUM function to a query to show the total of each salesperson's sales. If you wanted the query instead to return only those salespersons whose total sales exceeded their sales quota, you could apply a group qualification to produce that result. BI Query would first sum the sales for each salesperson, then filter for those sums that exceed the quota. You could not achieve this result if you restricted the query to a value of "greater than the sales quota", then applied the SUM function. In this case, you would be telling BI Query to restrict results to only those individual sales that by themselves exceeded the quota (if any); BI Query would apply the SUM function to these results.

Applying a Function After Grouping

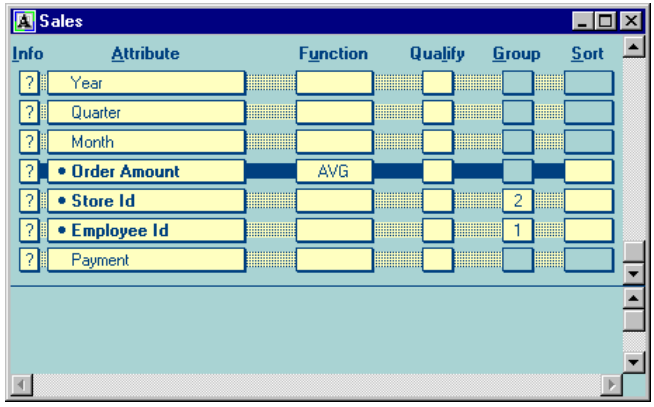
When you apply an aggregate function to an attribute, BI Query automatically groups all the other attributes in the query in the order in which you selected them. You can also specify your own group order by clicking the appropriate Group boxes in the attribute window. If you do not specify an order for each selected attribute, BI Query automatically orders them in the order in which you selected them.

To apply a function after grouping:

1. In an attribute window, select the attributes you want and apply any necessary qualifications.
2. Click the Function box for an attribute, then select a function from the pop-up menu that opens.



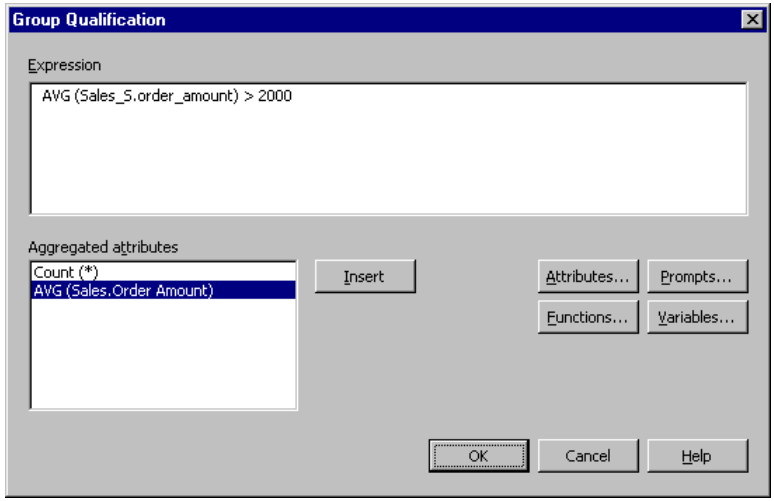
3 To group the data in a different order than the order in which you selected the attributes, click the Group box for the remaining attributes. (The function is applied after the results are grouped.)



This query calculates the average of all order amounts. Grouping occurs first for employee IDs, then store IDs

Applying a Function Before Grouping

Use the Group Qualification dialog box to build the expression for a group qualification.



Aggregated Attributes lists the Count function, plus any attributes to which an aggregate function has been applied and any calculated attributes that are treated as aggregates

To perform a calculation before grouping results:

1 Construct a query using an aggregate function.

2 On the Query menu, point to Qualification, then click Group. The Group Qualification dialog box opens.

3 From the Aggregated Attributes list, select the column with the aggregate function to which you want to apply the group qualification, then click Insert. The function and attribute appear in the Expression box.

For example, if you have applied the AVERAGE function to an order amount attribute, the Expression box might read the following:

```
AVG(Sales_S.order_amount)
```

4 Type the qualification in the Expression box.

For example, to find only averages greater than $2000, you would add "> 2000" so that the complete entry in the Expression box reads as follows:

```
AVG(Sales_S.order_amount) > 2000
```

If you had grouped before calculating, only amounts greater than $2000 would have been averaged.

You can also use the available buttons to build the expression:

- To include a prompt in the qualification, click Prompts, then use the Prompts dialog box to select and insert it.
- To include a variable, click Variables, then use the Variables dialog box to select and insert it.
- To include another attribute in the qualification, click Attributes. In the Insert Attribute Name dialog box, select the data object and attribute name, then click Insert.
- To include a built-in or DBMS-specific function, click Functions, then use the Functions dialog box to select and insert the function.

5 Click OK.

6 Submit the query. The group qualification will be retained with the current query unless you remove it from the Group Qualification dialog box or click New on the Query menu.

Note: When apply group qualifications to restrict the number and types of groups returned, you can specify a group order as well.

Grouping without Functions

To eliminate duplicates, you can group attributes without applying functions. For instance, if you select the Country attribute for a query, the query retrieves all names of countries in the database table, listing each in the order found and as many times as it is found. If you group by Country, your results set contains the name only once for each group. For example, no matter how many times Canada appeared as a country, the name "Canada" appears only once in the list, as the name for the whole group. You can use this method to produce results sets for data values results files.

For more information on the Distinct modifier, see "Using Query Modifiers" on page 43.

Grouping attributes in this way is similar to applying the Distinct modifier to eliminate duplicates. Some databases perform better if you group attributes rather than apply the Distinct modifier to the query.

> **Note:** When you group attributes, most DBMSs also sort the results (alphabetically, numerically, and chronologically). However, grouping attributes is not the same as sorting them, because the sort process does not remove duplicate rows. For more information on sorting, see "Sorting Rows" on page 31.

To group without applying a function to an attribute:

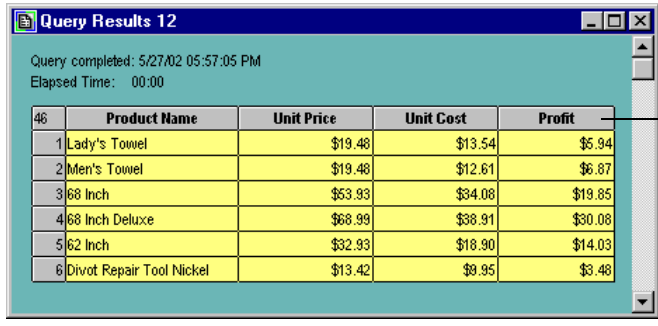
1 In an attribute window, select the attributes you want.

2 To group the data, click the Group box for the remaining attributes.

Creating Calculated Attributes

BI Query allows you to add calculated attributes to a data object. A calculated attribute performs a user-defined calculation in a query. The calculation can be based on more than one attribute from more than one data object in a design window. You can also apply a function to an attribute in the calculation.

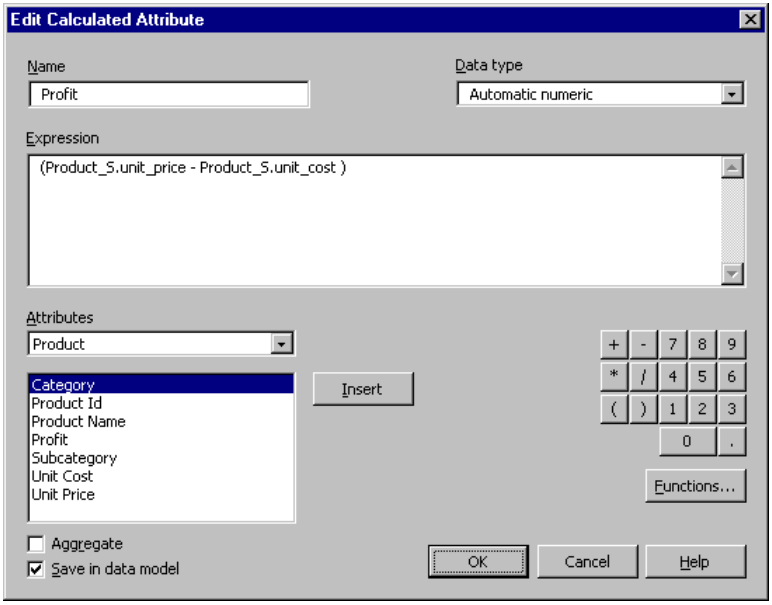
For example, a data object may include Unit Price and Unit Cost attributes that retrieve a product's market price and production cost, respectively. By creating a new attribute that calculates Unit Price minus Unit Cost, you can determine the profit made on each product.


Query Results 12
Query completed: 5/27/02 05:57:05 PM
Elapsed Time: 00:00
46
Product Name
Unit Price
Unit Cost
Profit
1 Lady's Towel $19.48 $13.54 $5.94
2 Men's Towel $19.48 $12.61 $6.87
3 68 Inch $53.93 $34.08 $19.85
4 68 Inch Deluxe $68.99 $38.91 $30.08
5 62 Inch $32.93 $18.90 $14.03
6 Divot Repair Tool Nickel $13.42 $9.95 $3.48
The calculated attribute Profit calculates the profit made on each product

For more information on editing and deleting calculated attributes, see the Help for your BI Query application.

In a combined data model, users can edit or delete any calculated attribute. In a split data model, users can edit or delete only calculated attributes that they create, not those provided by the administrator.

Use the Edit Calculated Attribute dialog box to create calculated attributes.



When you create a calculated attribute, you can either set a specific data type for the attribute or let BI Query automatically set the data type based on the calculated values.

> **Note:** If you select an automatic data type for a calculated attribute, BI Query cannot run the default `SELECT DISTINCT` query for the attribute. In this case, if you want to qualify the attribute, you cannot select the qualification value(s) from a list of data values retrieved from the database.

To create a calculated attribute:

1 Double-click the data object to which you want to add the calculated attribute. (BI Query must not be in Design Mode.)

2 In the attribute window, click the Function box for the attribute below which you want to add the calculated attribute.

3 On the pop-up menu that opens, click Calculation. The Edit Calculated Attribute dialog box opens.

4 In the Name dialog box, type the name you want to give to the calculated attribute.

For more information on available data types for calculated attributes, see the Help for your BI Query application.

5 From the Data Type drop-down list, select the data type for the attribute.

If you want the type to automatically match the data type of the calculation, select either Automatic Numeric or Automatic Character, depending on the calculation. You can also select a specific data type for the attribute.

> **Warning!** If you're connecting to a Teradata DBMS and you're not using ODBC, you must select a specific data type for the calculated attribute. Do not use the defaults Automatic Numeric or Automatic Character.

6 In the Expression box, type the calculation. Add a space between each item in the calculation. You can also build the calculation using the available buttons, as follows:

- To enter the database name of an attribute, click Attributes, then use the Insert Attribute Name dialog box to select and insert the name.
- To enter a number or arithmetical operator, use the key pad at the right of the dialog box.
- To insert a function into the qualifier, click Functions. In the Functions dialog box, select the function you want to insert from the Functions list or Other Functions list, then click Insert.

For more information on using functions in calculated attributes, see "Using Functions in Query Expressions" below.

If you select a built-in function from the Functions list, it automatically applies to the selected attribute; the function and the database name of the attribute appear in the Expression box and the Aggregate option is selected. (This ensures that when the calculated attribute is used in a query, the other attributes in the query are grouped.)

7 To ensure that the other attributes in the query are grouped when the calculated attribute is used in a query, select Aggregate.

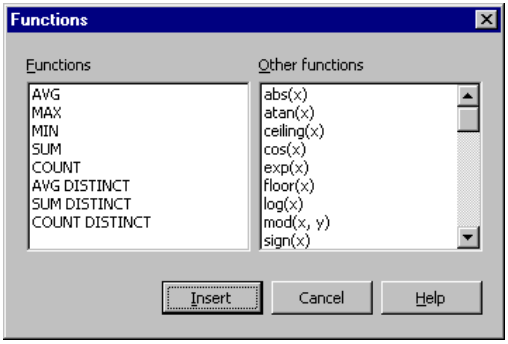
8 To save the attribute for future sessions, select Save in Data Model.

9 Click OK. The calculated attribute is added below the attribute you selected in step 2.

Using Functions in Query Expressions

Use the Functions dialog box to include a function in the following types of expression:

- the expression used in a group qualification
- the calculation for a calculated attribute
- the expression that defines a variable



The Functions dialog box displays a list of built-in functions in the Functions list and, for certain DBMSs, a list of scalar functions in the Other Functions list. (A scalar function takes a predefined number of parameters—usually one or two—and produces a value for each result in turn. Aggregate functions are often used with a GROUP BY clause; scalar functions aren't.)

You can use most scalar functions available in your DBMS even if they don't appear in the Other Functions list. Manually type the function as part of the expression. (Check your DBMS documentation for the correct format.)

When you run the query, BI Query evaluates the function and inserts its return value into the expression.

Using Subqueries

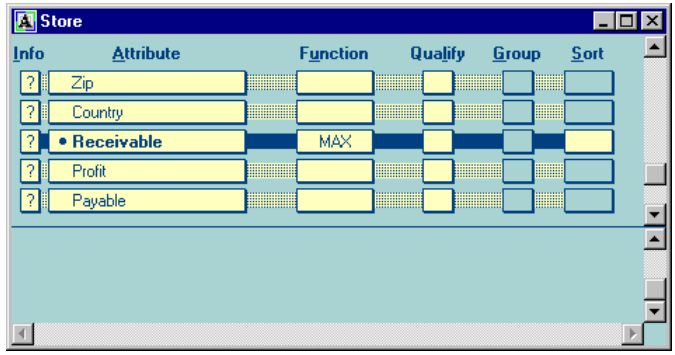
You can access your data in more dynamic ways by qualifying a query with a nested query statement, or subquery. A subquery returns values for a single attribute. A subquery doesn't have to qualify the attribute on which it's based, but it must have the same data type.

When you qualify an attribute with a subquery, you cannot include any other qualifiers on that attribute (such as other subqueries, data values results file, data sets, prompts, and so on). If you use a subquery to qualify an attribute that is already qualified with other data values, those values are removed from the qualification.

You cannot delete a subquery from a qualification without removing the entire qualification.

Example

To retrieve information about retailers with the highest credit limit, you could build and save a query that includes the Receivable attribute with the MAXIMUM function applied to it.



When you build other queries, you can insert that query as a subquery into a qualification to obtain results based on the maximum receivable amount. When you submit such a query, BI Query submits the subquery first, then uses the results to qualify the query. In this way, the query is always qualified with the current maximum value in the database.

Creating and Inserting Subqueries

For more information on subqueries, see the Help for your BI Query application.

To qualify a query with a subquery, you need to create the subquery first, then insert it into the qualification tree.

To create a subquery:

1 In the attribute window, click an attribute, then qualify it and apply any other necessary restrictions or modifiers.

> **Note:** Do not sort the attribute. Subqueries cannot supply sorted data values.

2 On the Query menu, click Save. The Save Query dialog box opens.

3 In the Query Name box, type a name for the subquery, then click Save.

To qualify an attribute with a subquery:

1 If necessary, click the New Query button on the Application toolbar to begin a new query.

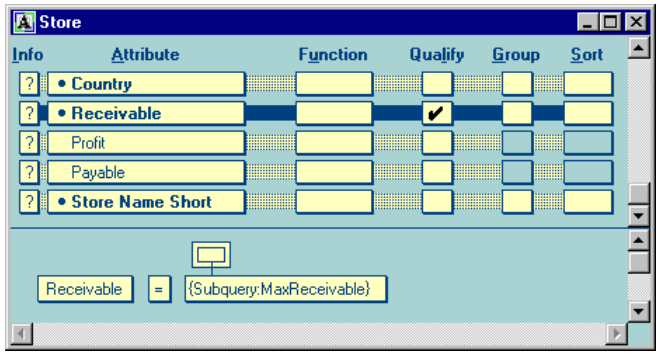
2 Double-click a data object, and then select the attributes for the query you want to build.

3 Click the Qualify box for the attribute you want to qualify, then click the Data Values button in the qualification tree.

4 On the menu that opens, click Subqueries.

5 In the Insert Subquery dialog box, select the subquery from the Queries list, then click Insert.

6 From the Operator drop-down list in the qualification tree, select an operator. (If the subquery is returning a single value, select =. If it is returning a list of values, select IN or NOT IN.)



7 Submit the query or save it for later submission.

Note: The Begins With, Contains, Ends With, Does Not Begin With, Does Not Contain, and Does Not End With operators are invalid for subqueries.

Using Data Sets

A data set is a text file that stores one or more data values. You can use data sets to qualify attributes in a query. Unlike data values results files, which you usually produce using BI Query, data set files can come from any source and can contain any values. For example, you can use an application like Microsoft Excel to create a file containing a series of comma-separated values.

When you include a data set in a query, all of the values stored in the data set qualify the query. Data sets let you include large blocks of values in a query without entering them one by one. They also let you edit those values using a text editor or spreadsheet program.

For more information on .qrr files, see "Naming and Storing Data Values Files" on page 80.

Like data values results files, data sets have an associated `.qrr` file. BI Query creates the `.qrr` file when you first include the data set in a query. You can use `.csv` (comma-separated) files and `.qrd` files (data values results files) as data sets.

Data Set File Syntax

Data set files have a specific syntax that you must follow if you create them in a text editor.

Distinct values must be delimited throughout the file by one of the following delimiters:

- a tab (ASCII 9)
- a comma (ASCII 44)
- a newline (ASCII 13 + 10)

Use only one type of delimiter throughout a given file. Make sure that the final value in the file is followed by a delimiter character.

Use the standard HTML/XML syntax for comments, as follows:

```
<!-- comment_text -->
```

where `<!--` and `-->` are literal strings as written and *`comment_text`* is the actual text of the comment. Data set comments appear on a line by themselves; a given comment cannot extend beyond one line.

Remember to store the data set file in the `DataVals` folder for the data model. When you publish data values files to the BI Server Repository, the data set files are also published. For more information on publishing data values files and data sets, see the *BI Query Data Models User's Guide*.

Example Data Set File

The following example of a data set file includes two comments and uses carriage returns to delimit data values:

```
<!-- N. American sales centers -->
Albany
Peterborough
Winnipeg
<!-- European sales centers -->
Bristol
Paris
Verona
```

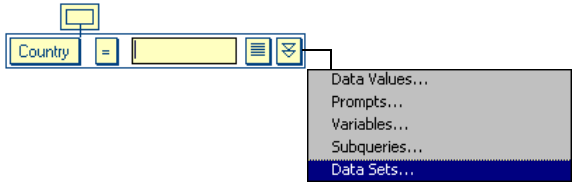
Using a Data Set to Qualify an Attribute

When you qualify an attribute with a data set, BI Query applies all of the values in the data set to the qualification. You can insert more than one data set in the qualification for a given attribute; in this case, all of the values in all of the files qualify the attribute. You can also combine data sets with other qualifiers, such as variables, prompts, and data values.

To qualify an attribute with a data set:

1 Click the Qualify box for the attribute.

2 In the qualification tree, click the Data Values button beside the attribute. On the menu that opens, click Data Sets.



3 In the Open Data Set dialog box, select the data set file that you want to apply to the attribute. Click Open. The qualification tree displays the name of the data set file in the form {Data set: *dsname*}, where *dsname* is the name of the file.

When the query is submitted, BI Query inserts the contents of the data set file into the qualification.

Viewing Data Sets

Once you have inserted a data set into a query, you can view the values it contains and the number of values in the set.

To view the contents of a data set:

1 In the qualification tree, click the List button beside the attribute that is qualified by the data set.

2 On the menu that opens, click View Data Set. (If you have applied more than one data set to the attribute, the Data Set dialog box opens; in this case, select the data set file from the list and click OK.)

The View Data Set dialog box opens. The top of the dialog box displays the name of the selected data set and the number of values it contains. The list in the center of the box displays the actual values.

3 Click OK to close the dialog box when you have finished viewing the values.

Using a Data Values Results File as a Data Set

You can use `.qrd` files as data sets, because they are essentially the same thing: a text file containing one or more delimited values. There are several advantages to using a `.qrd` file as a data set rather than as a data values file:

- All values in the `.qrd` file apply to a query. You do not need to select the values as a group. This makes it easy to apply a large set of values to a qualification.
- You can edit multiple values in the query by editing the file: you do not need to edit the SQL or the qualification tree to change the values.
- The values in the data set are applied to the qualification at run time; they are not saved with the query. This means that some other person or process can update the data set; any scheduled query that includes the data set will automatically use the updated qualification values.

For more information on scheduling a query, see "Using BI Server Scheduler" on page 249.

Using a `.qrd` file as a data set also lets you include values that contain delimiter characters (for example, `1,330.00`, which includes a comma). Each `.qrd` file automatically comes with a `.qrr` file that BI Query uses to interpret the data. If you create a data values results file for an attribute whose values contain delimiters, BI Query lets you choose other delimiters for the file.

For information on editing data values results files, see "Creating Data Values Results Files Using a Text Editor" on page 83.

Creating Tables in the Database from Query Results

Most of the time, the tables you need already exist in the database. However, any query that retrieves information from a number of tables takes time because it uses multiple joins. Summarizing the information in one table eliminates the joins and reduces the number of rows that the DBMS has to process. This gives you faster access to the information and reduces the load on the database server. For example, instead of querying the Stores, Sales and Sales Detail (three tables and two joins) tables to get a store's order history, you could create an order history table that combines those records together in one table.

If the database administrator grants permission to create tables, and if the Edit Data Model, Save Data Model, and Submit to Table permissions are assigned, you can create tables in the database from query results.

This method works best when you need to analyze information that doesn't change regularly. For example, by creating a table that summarizes sales for a particular year, you can query the table and perform calculations on the results without the values changing and becoming out of date.

A query that submits results to a table can be linked to a button so that the process of creating new tables in the database is automated.

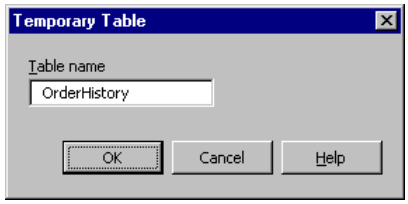
When results are saved to a table in the database, BI Query creates in the active design window a data object that corresponds to the table. If the Submit to Named Table permission is assigned, you can name the new table and the corresponding data object. Otherwise, BI Query generates the name.

For information on deleting user-defined data objects (and the corresponding tables), see the Help for your BI Query application.

To create a table in the database from query results:

1 Create a query that retrieves the information you want to include in the table.

2 On the Query menu, point to Submit To, then click Table. The Temporary Table dialog box opens.



3 Type a name for the new table, then click OK. When you save the data model, the new table is saved with it.

> **Warning!** If you have the Submit to Named Table permission assigned, this name is also applied to the corresponding table in the database. If this query is linked to a button, each time the button is used, BI Query overwrites the table with current results. Otherwise, BI Query creates a new table each time and allows the user to name only the corresponding data object.

Creating Dynamic Relationships

For more information on relationships, see "Data Model Components" on page 4.

The relationships between data objects in a data model let you form queries using more than one object. Rather than using all possible relationships between data objects, the data model typically provides only the most important ones. When data objects don't have a relationship, or when existing relationships don't relate the attributes the way you need them to be related, you can create dynamic relationships. Dynamic relationships are typically created "on the fly" for temporary use.

For more information on join conditions, see the *BI Query Data Models User's Guide*.

In order to create a dynamic relationship between two tables, you need to decide the condition under which the data in one table is related, or "joined," to data in the second. That is, you need to specify a join condition.

For example, you can create a relationship between the Retailer and Sales tables, where the Retailer Number attribute in one table is made equal to the Retailer Number attribute in the other table. A query that includes both data objects will then return data only for retailer numbers that appear in both tables.

Choosing a Join Operator

When you relate two data objects, you must specify an operator to determine the join condition. The available operators depend on the DBMS you are using.

The default operator, =, is the most common type of relationship; it specifies that the value of an attribute from one table equals the value of an attribute from another. You can also use other comparison operators (greater than, less than, greater than or equal to, less than or equal to) for the join condition. You may need to consult your administrator to ensure that the relationships you create use the appropriate join conditions and produce the results you want.

> **Note:** Depending on the DBMS you are using, you can specify outer joins in a dynamic relationship using the *= and =* operators or the new-syntax check boxes in the Choose Relationship dialog box. For more information on outer joins, see "Types of Column Joins" on page 144.

Basic Procedure

Creating dynamic relationships requires the Dynamic Relationships permission and the Allow Dynamic Relationships preference. Saving the relationships requires the Save Data Model permission.

You don't need permission to save a dynamic join as part of a query. When the query is reopened, the dynamic join is recreated temporarily for that query to use.

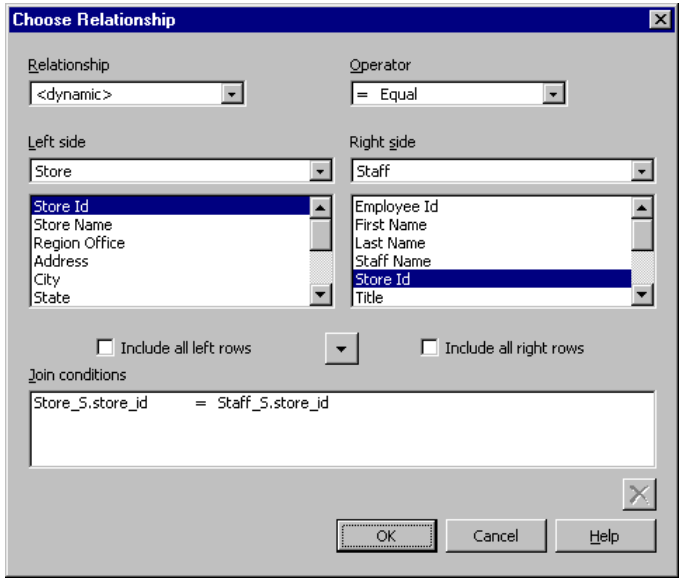
To create a dynamic relationship:

1 Open the Choose Relationship dialog box by doing one of the following:

- Select the two data objects you want to join (Shift + click). On the Query menu, click Relationship.
- Control + click and drag from one data object to the other.

2 In the Choose Relationship dialog box, from the Relationship drop-down list, select <dynamic>.

If no relationship currently exists between the selected data objects, <dynamic> appears by default. If a relationship already exists, the Join Conditions box displays the relationship and the list of attributes for the selected data objects is dimmed. You can create a dynamic relationship between any data objects, even if they are already related.



3 From the Left Side drop-down list, select a data object, then click an attribute in the list below Left Side.

4 From the Right Side drop-down list, select a data object, then click an attribute in the list below Right Side.

5 Specify the join type using the appropriate procedure:

Join Type	Procedure
Inner join	From the Operator drop-down list, select a join operator. Ensure that the Include All Left Rows and Include All Right Rows check boxes are cleared and that the join operator is not *= or =*.
Half Outer join (Classic syntax)	From the Operator drop-down list, do one of the following: • Select *= to make the Left Side data object row preserving. • Select =* to make the Right Side data object row preserving. The design window must support Classic syntax.

Join Type	Procedure
Any outer join (ODBC or ANSI SQL/92 syntax)	From the Operator drop-down list, select a join operator, then do any of the following: • To make the Left Side data object row preserving, select Include All Left Rows. • To make the Right Side data object row preserving, select Include All Right Rows.

Note: The Include All Left Rows and Include All Right Rows check boxes are available only in design windows that use either the ODBC or ANSI SQL/92 join syntax, and only if the design window doesn't contain a legacy outer-join operator.

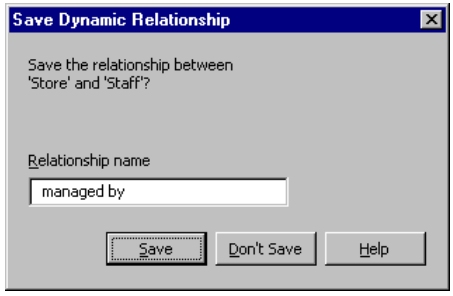
6 Click the down arrow button.

7 Click OK.

8 To save a relationship and name it, on the Query menu, click New.

9 In the BI Query dialog box, click Save.

10 In the Save Dynamic Relationship dialog box, specify a relationship name, and click Save. The name should reflect how the two data objects are related and should be unique.



Note: If you have created more than one relationship, repeat for each. To remove a relationship, click Don't Save in the Save Dynamic Relationships dialog box. When you save the data model, the relationship(s) are saved with it.

Deleting a User-Created Relationship

Users can delete relationships they create only if they have not already saved them.

To delete a relationship:

1 Do one of the following:

- Open another query.
- On the Query menu, click New.

The BI Query dialog box displays the message "Save dynamic relationships before deleting current query?"

2 In the BI Query dialog box, do one of the following:

- Click Delete to delete all new relationships.
- Click Save. In the Save Dynamic Relationships dialog box, click Don't Save for each relationship you want to delete.

Note: BI Query deletes the relationships that you have created but not saved. Relationships will still be part of the query if you have saved the query.

Using Outer Joins in a Query

For more information on outer joins, see "About Column Joins" on page 143.

For queries containing outer joins, the order of the joins can significantly affect the query results. For this reason, once you add an outer join to a query, BI Query sets an order for every join. If you have selected the Show Join Order preference, BI Query displays the join order for the connections (if the join order is relevant to the current query).

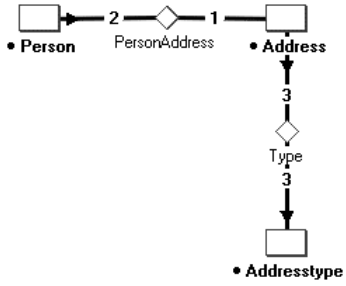
Note: There can be two or more joins with the same join order. This is because join order is based on the level of nesting. Two joins that are nested at the same level within other joins have the same join order.

For more information on merged data objects, see the *BI Query Data Models User's Guide*.

If there are merged data objects in the query, and the Automatically Order Outer Joins preference or permission is assigned, BI Query treats each join within the merged object as any other join and orders it automatically. If the preference and permission are not assigned, the join order for each join in the merged object matches the order in which the join was added to the object. If the Show Join Order preference is set, the ordinal number for joins on either side of a merged object will not likely be consecutive. (The missing ordinals belong to joins inside the merged object.)

Example 1

In the following example, the inner join between the actual relationship PersonAddress and the data object Address takes place first (1). Person is then left-outer-joined to the first join (2). The result is then left-outer-joined to AddressType (3).



The SQL (ANSI SQL/92) for this example is:

```
SELECT Person.*, Address.*, AddressType.*
FROM
(
     (
          Person LEFT OUTER JOIN
          (
               PersonAddress INNER JOIN Address
               ON Address.ID = PersonAddress.AddressID
          )
          ON Person.ID = PersonAddress.PersonID
     )
     LEFT OUTER JOIN AddressType
     ON Address.AddressTypeID = AddressType.ID
);
```

Example 2

In the following query, both of the inner joins have an order of '1', whereas the outer join has an order of '2':

```
SELECT A.*, B.*, C.*
FROM
(
     A INNER JOIN B ON A.x = B.x
)
LEFT OUTER JOIN
(
     B INNER JOIN C ON B.y = C.y
)
ON A.z = B.z;
```

This query doesn't tell the DBMS how to order the inner joins with respect to each other because they occupy the same level of nesting.

Specifying Join Order

You can define the order of joins either automatically or manually.

To specify join order automatically:

Select either the Automatically Order Outer Joins preference or the Automatically Order Outer Joins permission. (You can set permissions using BI Query Admin only.) BI Query automatically chooses the join order for you. When selected, the Automatically Order Outer Joins permission overrides the Automatically Order Outer Joins preference.

> **Note:** The join order that BI Query automatically determines for a given query depends on the order in which you add the joins to the query. However, different join orders may produce the same results set. If you deselect and reselect joins within a query, the automatic join order may change but the outcome will be the same.

To specify join order manually:

Clear the Automatically Order Outer Joins preference and permission. Select (click) the joins in the desired order.

If you aren't satisfied with a join order chosen manually, you can change it by deselecting the connection lines and then selecting them again in the order you want.

About Join Conflicts

A join conflict occurs when more than one join could supply NULLs to the same *materialized relation* (either a nested join that is performed in advance of the joins in question or an actual table).

BI Query cannot resolve join conflicts on its own; instead, it can highlight the conflicting joins. If you want BI Query to display join conflicts, select the Show Join Conflicts preference. Conflicting joins appear as dashed connection lines.

Join conflicts do not appear in BI Web.

Example

Consider three tables, A, B, and C, with the following data:

Table A

Name	ID
Jane Smith	100
Hector Alvarez	200
Ella Bruce	300
Francis Coulomb	400

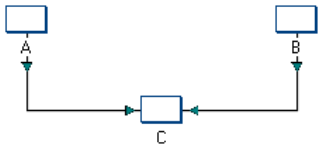
Table B

Country	CountryCode
UK	0114
Canada	2010
Mexico	2276
Japan	4531

Table C

ID	Title	Country
100	Manager	France
200	Writer	Mexico
250	Manager	Germany
500	Assistant	Australia

Tables A and B each form outer joins with C such that A and B are both row-preserving; C in this case is the materialized relation:

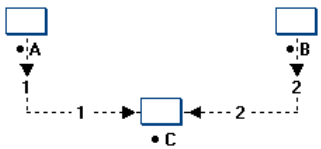


If the join between A and C has order 1, and you create a query involving both A and B, the join conditions between A and C are tested first. The results of this first join contain all rows from A, plus those from C that match A on the joined column (ID). Table A supplies NULLs for its unmatched rows:

Name	ID	Title	Country
Jane Smith	100	Manager	France
Hector Alvarez	200	Writer	Mexico
Ella Bruce	300	NULL	NULL
Francis Coulomb	400	NULL	NULL

The results of the second join contain all rows from B, plus those from the results of the first join that match B on the joined column (Country). However, the NULL values supplied by A do not match any values in B in the joined column, so that the corresponding rows from the first join are not part of the second set of join results. In other words, the final result does not preserve all rows from table A. If the join between B and C has order 1, a similar problem occurs (not all rows from B are preserved).

BI Query represents join conflicts using dashed connection lines:



About Outer-Join Syntax in Queries

For more information on outer-join syntax types, see the *BI Query Data Models User's Guide*.

BI Query supports the "Classic" (ANSI SQL/89) outer-join syntax that uses the legacy operators *= and =*. It also supports the "new" ODBC and ANSI SQL/92 types of outer-join syntax. The Classic and new types of syntax cannot appear in the same design window.

The particular outer-join syntax in use in a design window (or across the data model) depends on the connection assigned to that window (or model). If you want to use a particular type of outer-join syntax in a query, you do not have to edit the existing joins; instead, you can change the syntax for the connection. In this case, BI Query applies the specified syntax to all outer joins that use the connection.

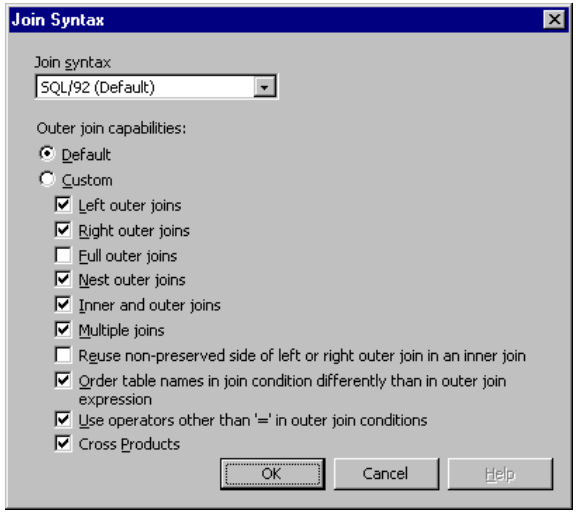
> **Note:** Because outer-join syntax information is specific to a given connection, it is stored in a connection file. When you change the syntax for a connection, it applies to all design windows and data models that use the corresponding connection file. If you want your changes to be local to a particular window or data model, create a new connection file, assign it to the window or model, then make the necessary changes to it. For more information on creating connection files, see the *BI Query Data Models User's Guide*.

Changing Outer-Join Syntax for a Connection

Use the Join Syntax dialog box to select an outer-join syntax for a connection file. The syntax you select must be supported by the connection.

To change the outer-join syntax for a connection:

1. On the Host menu, click Connections.
2. In the Connections dialog box, select the connection file you want to use, then click Edit.
3. In the Edit Connections dialog box, click Join Syntax. The Join Syntax dialog box opens.
4. From the Join Syntax drop-down list, select the outer-join syntax type that you want to apply to the connection. The dialog box displays the default settings for the selected syntax.



5 Click OK. The Join Syntax dialog box closes.

6 In the Edit Connection dialog box, click Save or Save As to save your changes.

Modifying the SQL for a Query

You can modify the SQL string generated by BI Query when you formulate a query. However, you can't directly modify the SQL for queries in which the results have been modified or combined with other results.

You must have the Edit Queries permission to modify the SQL. You also need the Show SQL String preference to display the SQL in the Query window. You can then modify anything after the SELECT word (subject to any restrictions placed on you by the DBMS).

BI Query expects results using the data format of the query selected in the design window. The modified query must return compatible results from the database. If it doesn't, the results may have missing, misaligned, or otherwise incorrect columns. To avoid this problem, do not change anything before the WHERE clause: don't change the attributes you've selected; don't delete, add, or change the function you've applied to them.

Changes you make to the SQL string should be the last changes you make to a query. If you modify the SQL string incorrectly, the DBMS may return an error message.

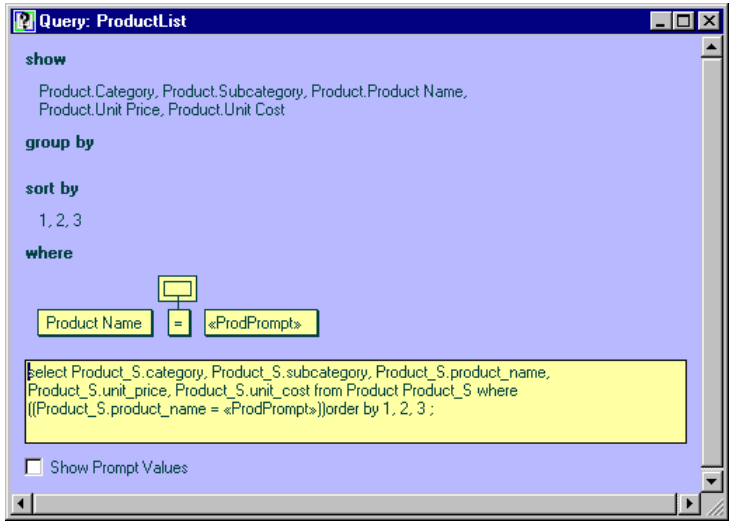
The SQL string in the Query window uses attribute names that appear in the database. When you edit the string, you must use these database names, not the display names that may appear in the attribute window.

To edit a standard query by modifying its SQL string:

1 Make sure that the Edit Queries permission is assigned and that the Show SQL String preference is set.

2 Create, open, or retrieve the query.

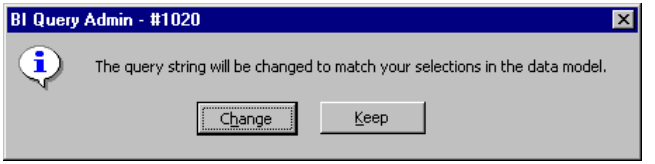
3 On the Query menu, click Show.



4 Make the changes to the SQL string.

> **Note:** If you select Show Prompt Values, BI Query displays in the SQL string the information provided for prompt qualifications the last time the query was run. You can't edit the SQL string while Show Prompt Values is selected.

If you subsequently modify the query in an attribute window after changing the query's SQL, BI Query displays an alert box to warn you that the modified SQL will be overwritten.



Do one of the following:

- Click Change to overwrite the SQL string you modified.
- Click Keep to undo your changes in the attribute window.

Using SQL Prefixes and Suffixes

You can use the Custom Check permission to improve the efficiency of the query. For more information, see the Help for your BI Query application.

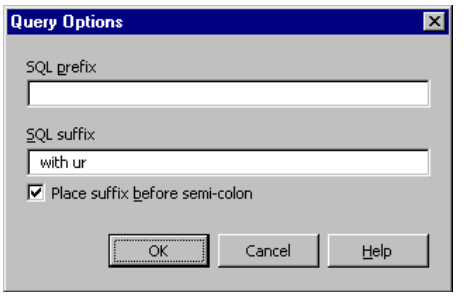
You can fine-tune the SQL you send to the DBMS by specifying a prefix and/or a suffix for every query and subquery that you submit.

For example, if you're using a DB2 DBMS on a VAX machine, you can improve database performance by appending the suffix `with ur` to every query to allow "uncommitted reads." If you're using Sybase, you can use the suffix `for browse`.

You can also specify whether the suffix occurs before or after the semicolon that terminates the SQL for queries. This feature requires the Edit Queries permission.

To specify a prefix and suffix:

1 In the Preferences dialog box, click Query Options.



2 In the Query Options dialog box, specify a prefix and/or a suffix.

3 To place the suffix after the semicolon that terminates the SQL, clear Place Suffix Before Semi-Colon.

4 Click OK.

Building Circular Queries

When the data model you're using contains two or more data objects whose relationships form a circular path, you can build queries that use all the data objects and all the relationships in that path. For example, if your data model contains the Employees and Departments data objects, and they're connected by the Work In and Managed By relationships, you can build queries that use both data objects and both relationships.

Because a data object in a circular path may be used in two roles, BI Query prompts you to specify whether or not to create a correlation name (or "alias") for the object. If you choose to do so, BI Query creates a temporary copy of the object using the correlation name.

You can also choose not to create a correlation. The choice you make determines how join conditions are applied in your query and the extent to which your results are restricted. A query that uses a correlation name allows you to ask two questions of the same data object and is less restrictive than a query that doesn't use a correlation.

Queries with a circular path are typically built by advanced users who understand how to use join conditions to retrieve the information they need.

Circular Queries with a Correlation

The last relationship you select when building a circular query determines which objects can be aliased: you can create a correlation for one of the two objects that form that last-selected relationship. The last-selected relationship also determines which join conditions apply to the correlation.

BI Query creates the correlation name for an object using the following format:

object__relationship

where

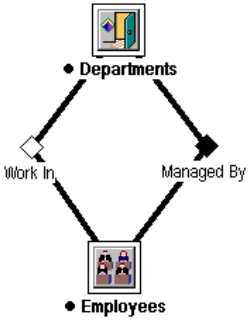
- *object* is the first five characters of the name of the aliased object
- __ is the literal string as written (two consecutive underscore characters)
- *relationship* is the first five letters of the last-selected relationship

For more information on correlation names, see the *BI Query Data Models User's Guide*.

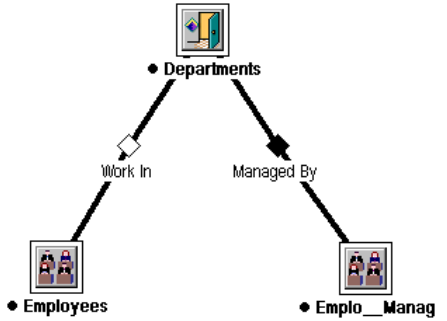
If you are using BI Query Admin, you can also use a correlation name to create a permanent copy of a data object. Users of the data model can then create a query involving the aliased data object more than once without creating a circular query at the same time.

Example

If you select the Managed By relationship last and create a correlation for Employees, BI Query creates a second, temporary copy of Employees called Emplo__Manag. The result is two copies of Employees: one connected to Departments through the Work In relationship, the other, temporary copy connected to Departments through Managed By.



This is what you see in the design window



Behind the scenes, BI Query creates a temporary copy of Employees and connects it to Departments using the Managed By relationship

When you double-click the original data object in the design window, you're prompted to specify which attribute window you want to use—the one for the original data object or the one for its correlation. You can use one attribute window to ask one question, then the other attribute window to ask the second question.

For example, suppose you want to use the data objects illustrated above to find out which employees work for managers who make over $60,000. You need a correlation because you're asking for two kinds of information from the Employees data object—the names of the employees and who is a manager making over $60,000. You'd use the Employees data object to find out the names of the employees, and you'd use its correlation to find out who is a manager making over $60,000. (You use the correlation for the second question because it's the one that uses the join condition in the Managed By relationship.)

When you submit the query, the DBMS processes it in two stages. Behind the scenes, the DBMS first obtains a preliminary results set (names of all employees and the departments they work in) by joining each row in Employees with each row in Departments, then applying the join conditions for Work In.

The DBMS then applies the join condition for Managed By to the preliminary results set and Emplo__Manag (the copy of Employees) to retrieve the final results (employees who work for managers making over $60,000).

The SQL for this query is as follows:

```
SELECT Employees.LastName
FROM Employees Emplo__Manag, Departments, Employees
WHERE
(
     (
          Emplo__Manag.Salary > 60000.00
     )
     AND Departments.DepartmentID = Employees.DepartmentID
     AND Departments.Manager = Emplo__Manag.EmployeeID
);
```

Building a Circular Query with a Correlation

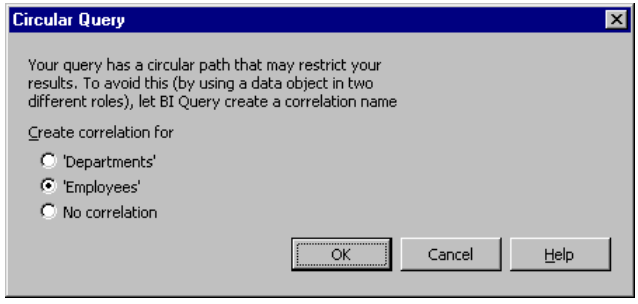
A circular query with a correlation is less restrictive than one without a correlation. Because circular queries can be complex, if you build one with a correlation, it's good practice to build it initially with an extra qualification to limit the number of rows returned. In this way, you can verify the design of the query without retrieving a possibly large amount of data.

To build a circular query with a correlation:

1 Decide which data object you want to create a correlation for and which relationship you want to use for that correlation.

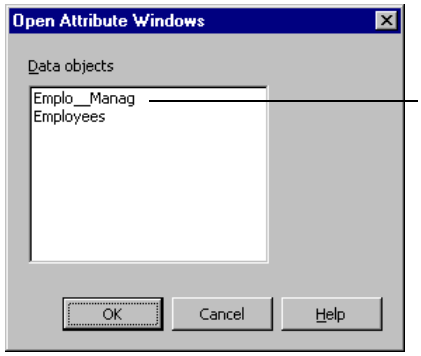
2 Click the relationships joining the data objects you want to include in the query, making sure that the last relationship you click is the one you want to use for the correlation.

3 In the Circular Query dialog box, click the data object you want to create a correlation for.


Circular Query
Your query has a circular path that may restrict your results. To avoid this (by using a data object in two different roles), let BI Query create a correlation name
Create correlation for
'Departments'
'Employees'
No correlation
OK
Cancel
Help

4 Click OK. In the design window, double-click the data object you created the correlation for.

5 In the Open Attribute Windows dialog box, click the attribute window(s) you want to open.



6 Click OK. In the attribute window(s), select the attributes you want to include in the query, then close the window(s).

7 Select any other attributes you want to use from the other data objects in the circular path.

8 To maintain all the join conditions in the query, on the Query menu, point to Modifiers, then deselect Trim Relationships (if necessary).

9 On the Query menu, click Run.

Circular Queries without a Correlation

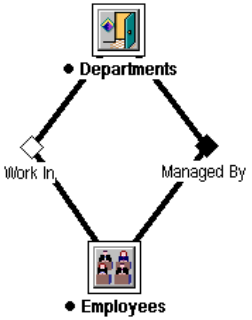
When you build a query without a correlation, BI Query doesn't create a temporary copy of a data object in the circular path. This allows you to ask one question using all the data objects and all the join conditions in the circular path. The results are usually more restricted because the rows returned from the database must meet all the join conditions in the circular path.

Example

Suppose you want to use the Employees and Departments data objects to find out which employees make over $60,000 and manage the department they work in. In this case, you need to create a circular query, but you don't need a correlation because you're using each data object only once.

When you submit the query, the DBMS processes it in two stages. Behind the scenes, the DBMS first obtains a preliminary results set (names of all employees who make over $60,000 and the departments they work in) by applying the join condition for Work In to Departments and Employees.

The DBMS then applies the join condition for Managed By to the preliminary results set to obtain the final results (employees who make over $60,000 and manage the department they work in).



Without a correlation, this circular query returns only the rows that meet the join conditions in both the Work In and Managed By relationships.

The SQL for this query is as follows:

```
SELECT Employees.LastName
FROM Departments, Employees, Departments
WHERE
(
     (
          Employees.Salary > 60000.00
     )
     AND Departments.DepartmentID = Employees.DepartmentID
     AND Departments.Manager = Employees.EmployeeID
);
```

Building a Circular Query without a Correlation

Building a query without a correlation is similar to building a query with a correlation except that the order in which you select the relationships isn't important. When BI Query prompts you to specify whether or not to create a correlation, you choose not to create one.

> **Note:** If the circular query contains an outer join, you must create a correlation. (The No Correlation check box in the Circular Query dialog box is dimmed.)

To build a query with a circular path that doesn't use a correlation:

1. Click the data objects and relationships you want to include in the query.
2. In the Circular Query dialog box, click No Correlation.
3. Click OK.
4. In the design window, double-click the data objects in the circular path.
5. In the attribute window(s), select the attributes you want to include in the query, then close the attribute window(s).
6. On the Query menu, click Run.

Chapter 11

Exporting Information

Exporting Information **219**

Preparing Results for Export **219**
Results Options and Separators 220
Specifying Results Options and Separators 222
Export Options 223
Setting Export Options 224

Export Methods **226**
Exporting Results to a Text File 226
Exporting Results to an Application Using a Button 227
Exporting Results Using E-Mail 229
Exporting the SQL for a Query 230

Exporting Results to Microsoft Excel **230**
Using Two DDE Commands 231
Opening Macro Sheets Automatically 231
DDE Command Examples for Excel 232
Combining Multiple Results Sets 233

Exporting Information

By exporting query results to other applications, you can extend the usefulness of your queries. BI Query exports results as text files that can be used by any application. For example, you can augment a report prepared in Microsoft Word by including actual data that you've retrieved from a query, or you can transfer results to a worksheet in Microsoft Excel to perform advanced statistical analyses. In addition to exporting query results directly to text files, you can do the following:

- export the underlying SQL for queries that have not been filtered or combined
- create buttons to export results automatically using menu commands or Microsoft's Dynamic Data Exchange (DDE)
- e-mail the query SQL files and results files that you've created
- copy the results from a result set in BI Query and paste them directly into a document in another application

If the appropriate permissions are assigned, you can also send the results of a query to a new table in the database. For more information on this procedure, see "Creating Tables in the Database from Query Results" on page 196.

Preparing Results for Export

Before exporting, you need to prepare your results for export. There are three things to consider:

- Display format—How numbers, characters, and dates are represented. For example, Microsoft Excel doesn't accept some date formats allowed in BI Query, like YY/MM/DD. For more information on data formats, see "About Data Formats" on page 57.

- Results options—Settings required by the receiving application. For example, a worksheet application may require double quotation marks around textual data. (If you're not sure what the receiving application requires, check its documentation.) For more information, see "Specifying Results Options and Separators" below.
- Export options—The receiving application, the export mechanism, and the commands to be executed. For more information, see "Setting Export Options" on page 224.

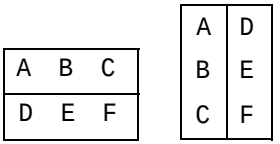
Results Options and Separators

When you save results to files or export them to other applications, be sure to specify the results options and separator characters required by the receiving application. You should also set these options if you are importing data to update the database.

Results Options

The results options you can set are as follows:

Results Option	Description
Save Query Name	Includes the name of the query that generated the results. This option is useful if you export to applications that use macros to perform certain operations depending on which query generated the data.
Save Column Headings	Includes results column headings.
Save Prompts	Includes the number of prompts used in the query (followed by a record separator), plus the following information for each prompt: • the prompt ID (followed by a field separator) • the number of values used in the prompt (followed by a field separator) • the prompt values (each separated by field separators)
Text in Quotes	Inserts double quotation marks around textual data.

Results Option	Description
Transpose	Transposes rows and columns so that the data is stored by columns rather than rows (for example, the first row of results becomes the first column of data).  Original Transposed This option applies only to results saved to a file (not results exported to an application). It is not available for results generated by buttons that link to queries.

Separators

Separator characters delimit each value in a row (normally a tab character) and each row of data (normally a carriage return). A separator can also indicate the end of the file.

A separator must be a valid ASCII character.

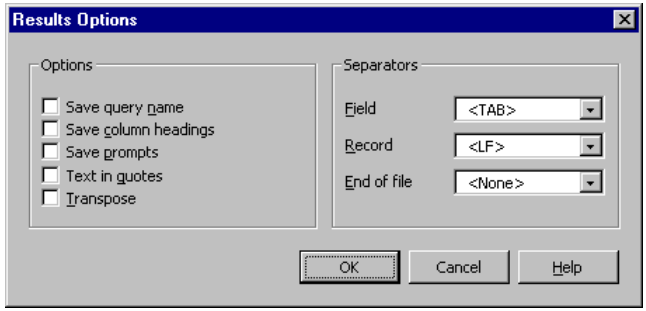
Results Options Targets

You can specify results options and separators for the following targets:

- The entire data model. In this case, the options you specify apply to all results sets and buttons you subsequently create, and to all data you import for updating the database. These options are the default for the model.
- A particular results set. In this case, the options apply only to the active results set; they override the data model defaults.
- The results generated by a button linked to a query. In this case, the options apply only to the results sets created when you click the button; the options override the data model defaults.

Specifying Results Options and Separators

Use the Results Options dialog box to set the options and separators for a particular results set, a button, or the entire data model.



To specify results options and separators:

1 Open the Results Options dialog box by doing one of the following:

- To specify default results options for the data model, on the Tools menu, click Preferences. In the Preferences dialog box, click Results Options.
- To specify results options for a particular results set, on the Results menus, click Options. (The results set must be active.)
- To specify results options for a button linked to a query, open the Edit Button dialog box for the button, select File from the drop-down list in the Output To area, then click Results Options.

For more information on options and separators, see the previous topic.

2 In the Results Options dialog box, specify the option(s) you want.

3 In the Separators area, specify the separator(s) you want from the Field, Record, and End of File drop-down lists. You can either type the separators or select them from the lists. Use the caret character (^) to represent the Control key. For example, type ^D to represent Control + D (ASCII 4).

BI Query provides the following codes to represent other ASCII characters:

Code	Description
<TAB>	The tab character (ASCII 9).
<LF>	The line feed character (ASCII 10).
<CR>	The carriage return character (ASCII 13).
<Space>	The space character (ASCII 32).

4 Click OK.

Export Options

If you create a button that exports results directly to another application, you need to specify the following export options:

- the application to which the results will be exported
- the export mechanism you're using (DDE or menu commands)
- the commands to be executed in the receiving application

When you export results to another application, BI Query copies the results, then checks whether the receiving application is already running and, if not, starts it. Depending on the export mechanism, BI Query then calls the appropriate menu commands in the receiving application or sends the DDE commands to the receiving application.

DDE lets you pre-program commands in the target application's programming language (such as Excel's macro facility) and further manipulate the results. For example, you could open an existing Excel spread-sheet and run a macro to perform a calculation not available in BI Query.

Whether you use menu commands or DDE depends to some extent on what you want the receiving application to do.

To Do This	Use This
Export results by specifying the menu commands to be executed in the receiving application. For example, to paste the contents of the Clipboard into a Microsoft Excel worksheet, you could specify the Paste command on the Edit menu in Excel.	Menu Commands
Include arguments in your commands that instruct the receiving application to carry out more specific tasks than simply using menu commands—for example, to open a particular worksheet and run a macro.	DDE

To determine the requirements of the receiving application, see its user documentation. You may also need the help of your administrator to find out whether an application supports DDE and what service name to use.

Results must be set up properly in order to be exported directly to an application. For more information, see "Specifying Results Options and Separators" on page 222.

Setting Export Options

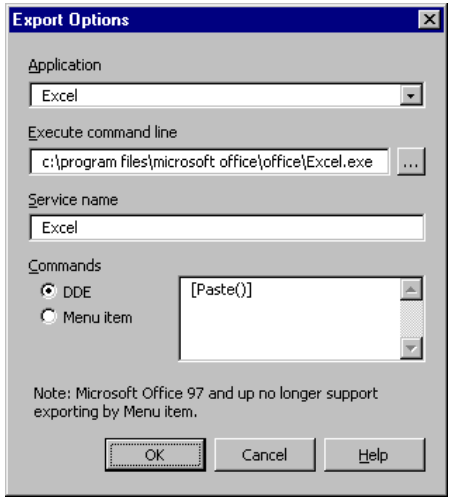
You can specify export options as a preference or when you create a button. Specifying export options in the Preferences dialog box applies the options to buttons created thereafter. Export options specified from the Create Button or Edit Button dialog boxes override the options specified as preferences.

To set export options:

1 Open the Export Options dialog box using one of the following methods:

- To specify default export options, open the Preferences dialog box (on the Tools menu, click Preferences). In the Preferences dialog box, click Export Options.
- To specify export options for a button, open the Create Button or Edit Button dialog box. In the Output To area, select Application from the drop-down list, then click Export Options.

2 In the Export Options dialog box, from the Application drop-down list box, select the application to which you want to export. If the application isn't in the list, select User Defined.



3 Do one of the following:

- If the path for the application is defined in your system PATH variable, type the name of the application in the Execute Command Line box. Use the name that appears in the title bar of the main window for the application.
- If the executable for the application isn't in your system PATH variable, type the full path and file name of the executable file in the Execute Command Line box.

Alternatively, click the browse button to the right of the Execute Command Line box. In the Select Application dialog box, browse to and select the executable file, then click Open.

4 If you want to paste results to a specific document, type the full path and file name of the document in the Execute Command Line box (after the application file).

5 If you're exporting using DDE, and the service name is not the same as the executable name, type the service name in the Service Name box. (The service name is the name that the application registers with the DDE server.)

6 In the Commands area, select DDE or Menu Item.

7 In the box to the right of the Commands area, type the commands to be executed in the receiving application.

For examples of DDE commands for Excel, see "DDE Command Examples for Excel" on page 232.

To send two DDE commands from BI Query, add a vertical bar (|) between the DDE command sequences you type. The first sequence is sent to the application only if it is not already running; BI Query must start it. You must enter all arguments in each DDE command.

The menu command mechanism requires the following format:

```
menucommand|command
```

For example, to run the Paste command on the Edit menu, type `Edit|Paste`.

8 Click OK.

Export Methods

When you export query results, you can either save the results or SQL to a text file, or send the results directly to an application using a button.

Exporting Results to a Text File

You can export query results to a text file in order to use them again in BI Query or retrieve them in other applications such as word processors or spreadsheet programs.

When you open results in BI Query, only the .qrd file appears in the Open Results File dialog box.

BI Query saves results as two files: one with the extension `.qrd` (ASCII text format for use in other applications) and one with the extension `.qrr` (BI Query-specific information, such as formats for each column and the SQL used to retrieve the results).

To save the results of a query:

1 Specify the results options required by the external application. For more information, see "Results Options and Separators" on page 220.

2 Open the Save Query Result to File dialog box by doing one of the following:

- Create or open the query that generates the results you want to save. On the Query menu, point to Submit To, then click File.
- With the results window active, point to Save As on the Results menu, then click Results.

3 In the Save Query Result to File dialog box, type a name into the File Name box. BI Query displays the corresponding file names with the .qrd extension.

4 Click Save.

5 In the external application, open the .qrd file to view the saved results.

When you delete an exported results set, be sure to delete both the .qrd and the .qrr files.

Exporting Results to an Application Using a Button

For more information on using buttons, see the *BI Query Data Models User's Guide*.

If the Edit Data Model and Save Data Model permissions are assigned, you can create a button that sends query results to an application. (To check, on the Tools menu, click Permissions.)

When you click the button, BI Query runs the linked query, formats the results according to the results options you have specified, copies the results to the Clipboard, and starts the target application (if it is not already running). If you have specified any export options—such as opening a particular document and pasting the Clipboard contents into it—BI Query also executes those commands.

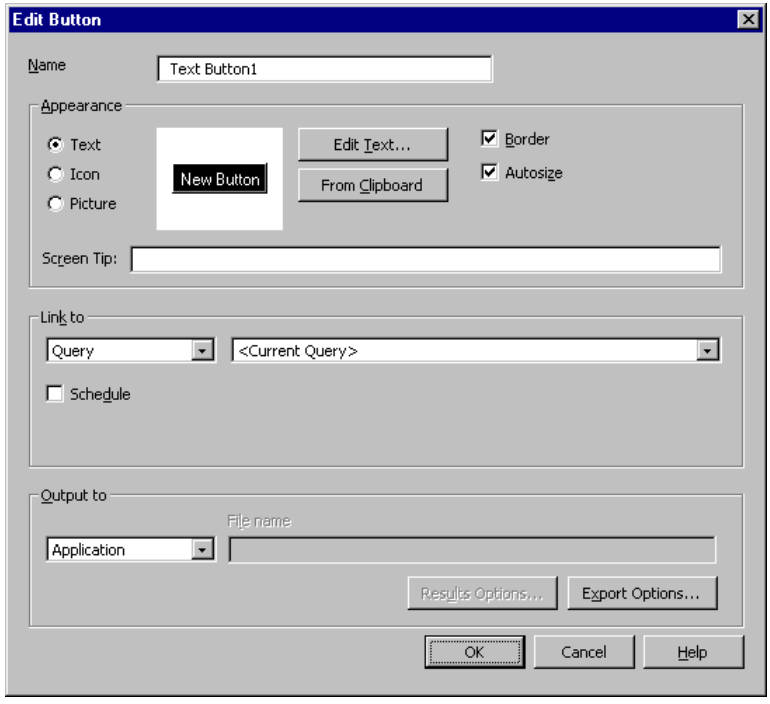
To use a button to export results to an application:

1 Build and save the query (if necessary).

2 If BI Query is not in Design Mode, click the Design Mode button on the Application toolbar.

3 On the Drawing toolbar, click the Button button, then click and drag the mouse over the area where you want the button to appear.

4 Double-click the button that appears. The Edit Button dialog box opens.



5 In the Link To area of the Edit Button or Create Button dialog box, select Query from the drop-down list box on the left, then select a query from the drop-down list on the right. (To submit any ad hoc query without creating a button for each, select <Current Query>.)

6 In the Output To area of the Create Button or Edit Button dialog box, select Application, then click Export Options. The Export Options dialog box opens.

7 Set the export options. For more information, see "Setting Export Options" on page 224.

8 Click OK. In the Edit Button dialog box, click OK.

Exporting Results Using E-Mail

For more information on using your e-mail client, see your client documentation.

If you're running an e-mail application on your computer, you can export BI Query files as attachments with your e-mail messages. E-mail provides an easy way to distribute new and revised queries and results to others. As long as your e-mail application supports Microsoft's Messaging API (MAPI), you can e-mail files directly from BI Query.

When you send e-mail from within BI Query, it starts your e-mail client and automatically attaches the query, results, or report files to the mail message, depending on the context in which you open the client:

- If you open or build a query before starting your e-mail client, BI Query automatically attaches the corresponding `.qry` file in your message—regardless of which window you're working in.
- If you open, retrieve, or generate a results set, then start your e-mail client, the corresponding `.qrd` and `.qrr` files are attached only if the results window is the active window. To automatically attach a results set, display the results window before starting your e-mail client.

For more information on standard report files, see "About Standard Reports" on page 151.

- If you open or save a standard report specification before starting your e-mail client, BI Query automatically attaches files corresponding to the current report (`.rpt`).

If you don't want to include the current query or results set in an e-mail message, remove the file from the enclosure.

To e-mail files:

1. On the File menu, click Send. Your e-mail application dialog box appears with a list of the enclosed files.
2. Remove any files you don't want to send (or enclose additional files).
3. Supply a subject for the message and the address of the recipient, then send the message.

> **Note:** When you retrieve e-mail that has been sent to you by another user, store query and report files in the `Queries` folder so that they appear in the corresponding directory dialogs when you open them. Store results in your main BI Query folder. Be sure you retrieve both `.qrd` and `.qrr` results files. To use retrieved files, close the data model if it's open, reopen it, then open the files.

Exporting the SQL for a Query

If the Save Queries permission is assigned, you can export the SQL for a query to a text file, provided that the query does not manipulate the results. The SQL can then be used in other applications. For instance, you can use the saved SQL in stored procedures in the database.

When you export a query from a results window, BI Query saves the query that generated that results set. When you export from any other window, BI Query saves the current query.

To export a query's SQL to a text file:

1 Open, create, or run the query you want to export.

2 On the Query menu, click Save.

3 In the Save Query dialog box, select Export SQL Only, then click Export.

4 In the Export Query dialog box, specify the location and the name for the query SQL text file, then click Save.

Note: You cannot export a separate SQL text file for a super query.

Exporting Results to Microsoft Excel

If you create a button that sends query results to Excel, the default DDE command is [Paste()]. When you export your results, BI Query starts up Excel (if it isn't already running), then instructs it to paste the contents of the Clipboard (the query results) into a worksheet.

Instead of using the default [Paste()] command, you can substitute up to two commands. For example, you could specify the commands to open a specific worksheet and paste results into it. Alternatively, if you are content to use a generic worksheet, you can open a macro sheet, run a macro on that sheet, then save the sheet under a different name.

To run a macro that is not on the global sheet, you must first open its sheet. However, you can configure a worksheet so that it opens the macro sheet automatically. You can then use two commands—opening a specific worksheet and running a macro—to obtain the effect of three.

Using Two DDE Commands

When you create a button that sends query results to Excel, you can specify up to two DDE commands. If the first of the two commands is to be executed only once (such as opening a specific worksheet on startup), use the pipe symbol (|) between the two commands:

```
[open(.........)]|[run(.............)]
```

(The pipe symbol has a different purpose for DDE commands than it does for menu commands.)

You can also use the pipe symbol to prevent Excel from doing either of the following:

- opening a worksheet that is already open
- pasting results into an open worksheet and overwriting the contents

If you use the pipe symbol, the first command is sent only if the application isn't already running and BI Query must start it up; the second command is always sent, even when the application is already running. Remember to specify all arguments for each command.

> **Note:** The pipe symbol is not necessary if it's logical for both commands to run each time you export results.

Opening Macro Sheets Automatically

You can extend DDE functionality by issuing two commands to obtain the effect of three. For example, when you open a specific Microsoft Excel worksheet and run a macro, you're doing three things—opening a specific worksheet, opening a specific macro sheet, and running a macro. The two commands would take the following form:

```
[open("c:\msoffice\excel\test.xls")]|[run("test.xlm!macro1")]
```

To obtain this effect, you must attach the Excel auto_open function to your worksheet so that your macro sheet opens whenever the worksheet opens.

To open macro sheets automatically:

1 Open the worksheet.

2 On the Formula menu, click Define Name.

3 Specify a name such as Auto_open, then use the following syntax in the Refers To box:

```
=test.xlm!open_macro
```

You can also achieve the same effect using either of the following methods:

- Place the macro on the global macro sheet (`global.xlm`). This sheet is opened automatically with Excel and doesn't require a command. Using it cancels opening a default worksheet, but that function can be included in the macro if needed. (The disadvantage to this approach is potentially developing a global macro sheet containing a lot of specific macros that aren't useful in every worksheet.)
- Place the macro sheet in your system PATH variable or in the subdirectory of the data model you're using. This is somewhat unorthodox, but the effect is the same as placing the macros on the global macro sheet, and it has the advantage of grouping the macros in a separate sheet.

DDE Command Examples for Excel

The following table lists common DDE command sequences that you can use to send results to Microsoft Excel. For other commands, see your Excel user documentation.

Command Sequence	Description
`[paste()]`	Pastes the results into a worksheet already open in Excel (or, if Excel is not already running, runs Excel, opens a new worksheet, and pastes the Clipboard contents into it).

Command Sequence	Description
`[new()]|[paste()]`	Pastes the results into a new worksheet in Excel.
`[open("c:\mydocs\test.xls")]|[paste()]`	Opens the specified worksheet (if Excel isn't already running) and pastes the results into it. If you often use the same format for a worksheet, you can set up a worksheet template, then use this command sequence to paste results into it. Once the results have been pasted, you can save the worksheet under a different name.
`[open("c:\mydocs\test.xlm")]|[run("test.xlm!macro1")]`	Opens the specified macro sheet, then runs the `macro1` macro on that sheet.
`[open("c:\mydocs\test.xls")]|[run("test.xlm!macro1")]`	Opens the `test.xls` worksheet, opens the `test.xlm` macro sheet, and runs the `macro1` macro. For this sequence to work, one of the following must be true: • The worksheet opens the macro sheet. • The macro sheet is in the same folder as the data model. • The macro is in the global macro sheet. For more information, see "Opening Macro Sheets Automatically" on page 231.
`[run("test.xlm!macro1")]`	Opens the `test.xlm` macro sheet, then runs the macro1 macro on that sheet. In this case, either the macro sheet must be in the same folder as the data model or the macro must be in the global macro sheet.

Combining Multiple Results Sets

You can combine the results of more than one query in an Microsoft Excel worksheet. This is useful when you're using data from different tables or when you're using a combined query. There are three methods you can use to combine multiple results sets in a worksheet.

Method 1

For more information, see "Exporting Results to an Application Using a Button" on page 227.

1 In BI Query, create a button for each results set. Link each button to the appropriate query. Set each button to send the query results to a macro that positions the cursor in Excel, then pastes the results into the appropriate location in the default worksheet.

   For example, to combine three results sets, you could set up three buttons to execute the following commands:

Button	Command Sequence
Button A	`[open("c:\msoffice\excel\test.xlm")]` `[run("test.xlm!macro1")]`
Button B	`[run("test.xlm!macro2")]`
Button C	`[run("test.xlm!macro3")]`

For more information on the global macro sheet, see "Opening Macro Sheets Automatically" on page 231.

   In this case, you would have to click button A first. However, if the macro sheet is in the BI Query data model folder, or if the macros are on the global macro sheet, you can accomplish the same result using the following commands:

Button	Command
Button A	`[run("test.xlm!macro1")]`
Button B	`[run("test.xlm!macro2")]`
Button C	`[run("test.xlm!macro3")]`

2 Click each button in turn.

   Don't close Excel until all three buttons have sent their results; otherwise, you'll create three separate worksheets.

Method 2

Create a button that selects the current query, then exports the results using the default paste to Excel. This method reduces the number of buttons but requires you to open each query in BI Query before running it. You must also reposition the cursor in Excel each time so that the next set of results does not overlap.

Method 3

Paste into a specific worksheet using the following procedure:

For more information on macro sheets, see "Opening Macro Sheets Automatically" on page 231.

1 Configure the worksheet so that it automatically opens the macro sheet (or use an alternative method to make the macros automatically available).

2 Create a button for each results set, then link each one to the appropriate query. Set each button to send the query results to a particular macro. For example, to combine three results sets, you could set up three buttons to execute the following commands:

Button	Command Sequence
Button A	`[open("c:\msoffice\excel\test.xls")]` `[run("test.xlm!macro1")]`
Button B	`[run("test.xlm!macro2")]`
Button C	`[run("test.xlm!macro3")]`

In this case, you must click button A first to open the worksheet.

3 Click each button in turn.

Chapter 12

Working with BI Server

Working with BI Server **239**
Requirements for Publishing Queries 239
Publishing Queries 239
Retrieving Queries 241
Deleting Published Queries 241
Publishing Standard Report Specifications 241
Publishing Data Values Results Files 243

Controlling Access Using BI Server **243**
Users and Groups in BI Server 244
Access Inheritance Rules 244
About the Set Security Dialog Box 245
Setting Security 246
Setting Security for Queries 248

Using BI Server Scheduler **249**
Scheduling a Query 249
Retrieving Scheduled Results 250

Working with BI Server

If you are in a BI Server environment, you can take advantage of BI Server's publishing, retrieving, scheduling, and security features. The requirements for using these features depend on the item you want to publish or otherwise access.

Requirements for Publishing Queries

For more information on scheduling queries, see "Using BI Server Scheduler" on page 249.

You can include queries when you publish and secure data models. You can also publish and secure queries individually. Published queries are available to other users and can be scheduled.

Although you don't have to save a query before you can publish it, the data model on which the query is based must have been published. If it hasn't, BI Query requires you to publish it before you can publish a query based on it. In order to be published, a query must also be the current query.

To publish a query to the BI Server Repository, you must also have at least one of the following BI Server system permissions:

- Publish Queries to Admin Layer
- Publish Queries to User Layer

In addition to queries, you can also publish data values results files.

Publishing Queries

When you publish a query, it's automatically saved locally, in the `Queries` folder where the data model is stored. (Data values queries are stored in the `DataVals` folder.) Then it's published to the Repository.

If you want to publish a query as a data values query file for a particular attribute, name the file using the following format:

`data_object.attribute`

where

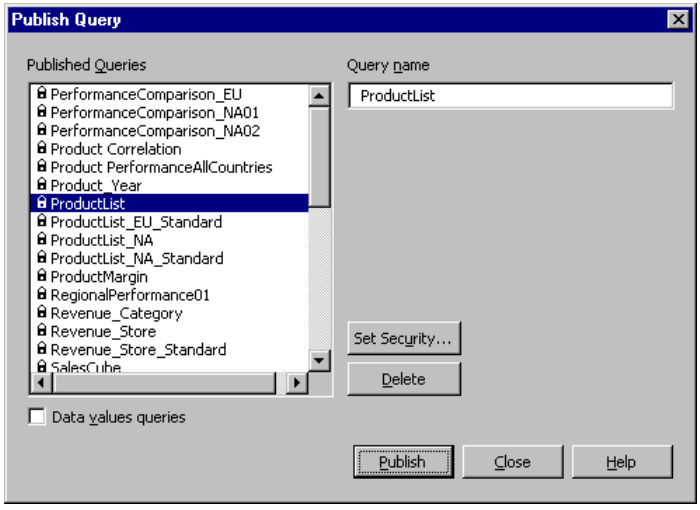
- *`attribute`* is the name of the attribute
- *`data_object`* is the name of the data object that stores that attribute

To make the data values available only for a specific prompt, name the file after the prompt. Attribute, prompt, and data object names are case sensitive.

To publish a query to the Repository

1 In a saved and published data model, create or open a query.

2 On the Query menu, click Publish. The Publish Query dialog box opens.



3 If you want to publish the query as a data values query file, select Data Values Queries.

4 In the Query Name box, type the file name for the query.

5 To set security for the query, click Set Security, then use the Set Security dialog box to grant or deny access to the query. For more information, see "Setting Security for Queries" on page 248.

6 Click Publish.

7 Test the query to be sure that it's been published. (Close it, then retrieve it from the Repository.)

Retrieving Queries

Before you can retrieve a query from the BI Server Repository, you must either open a local copy of the published data model on which the query is based or retrieve the published data model from the Repository.

To retrieve a query from BI Server:

1 On the Query menu, click Retrieve.

2 In the Retrieve Query dialog box, select the query you want, then do one of the following:

- To retrieve the selected query, click Retrieve.
- To retrieve and submit the query, click Run.

Deleting Published Queries

When you delete a published query from the Repository, you can delete the local copy at the same time.

To delete a published query:

1 In a saved and published data model, on the Query menu, click Retrieve.

2 In the Retrieve Query dialog box, select a query, then click Delete. The Delete Published Query dialog box opens.

3 If you want to delete the local copy of the selected query at the same time, select Also Delete Local Copy.

4 Click Yes.

Publishing Standard Report Specifications

If you are in a BI Server environment, you can publish a standard report specification to the Repository, where it will be available to BI Web users and other BI Query users. By default, the corresponding query for the report is also published. You cannot publish a standard report to the Repository unless the corresponding data model has also been published.

BI Web users can open a standard report only if the following conditions are true:

- the report specification has the same name as the corresponding query
- the query has been published
- the users have been granted access to the query

Published standard report specifications inherit the security setting applied to the data model. You cannot set security on a specification independently of the model. However, you can effectively grant or deny access to a standard report by granting or denying access to its query, because a specification is visible to BI Web users only if its query is also visible. Similarly, BI Query users who have retrieved a data model can refresh its standard reports only if they have been granted access to the corresponding queries.

To publish a report specification:

1. Create and format the report. Add any required calculations to it.
2. On the Report menu, click Publish Report Specification. The Publish Report Specification dialog box opens.
3. In the Report Name box, type the name for the specification you want to publish. Do not include the `.rpt` extension.
4. If you do not want to publish the corresponding query with the specification, clear Publish Query Also.
5. To control access to the specification, click Set Security, then use the Set Security dialog box to grant or deny access to the corresponding query. The Set Security button is available only if Publish Query Also is selected.
6. Click Publish.

For more information on security, see "Setting Security" on page 246.

Published standard reports appear in the BI Web Portfolio in a special subfolder for each data model. When a BI Web user opens a standard report in the Portfolio, the report is automatically refreshed.

Publishing Data Values Results Files

For information on retrieving and deleting published data values files, see the Help for your BI Query application.

When you publish a data values results file, you can make its data values available in qualifications and prompts for a particular attribute. To do so, to name the file using the following format:

data_object.*attribute*

where *attribute* is the name of the attribute and *data_object* is the name of the data object that stores that attribute.

To make the data values available only for a specific prompt, name the file after the prompt. Attribute, prompt, and data object names are case sensitive.

To publish a data values results file:

1 Create a data values results file or open an existing one.

2 On the Results menu, click Publish Data Values. The Publish Data Values Results dialog box opens.

3 In the Data Values Results Name box, type the name of the file. BI Query automatically adds the extension .qrd to the file when you publish it.

4 Click Publish.

Controlling Access Using BI Server

For any item you publish to the BI Server Repository, there may be people in your organization who need access to the information contained in it, people who need customized views of it, and people who shouldn't see it at all. You must secure the data so that it can be viewed or changed only by the people with the authority to do so.

The Set Security dialog box lets you control individual as well as group access to the information you publish; you can grant or deny general access and provide individual exceptions as necessary. You exercise this control based on a structure of users and groups that the BI Server administrator creates.

Depending upon your needs, the administrator may make it possible for you to grant system permissions to other users, such as the ability to schedule queries and reports. For information on these permissions and on how they're assigned, see the *BI Server Administrator's Guide*.

Users and Groups in BI Server

Your BI Server logon user name and password identify you as a user in a structure of users and groups created by your BI Server administrator. The administrator assigns system permissions that determine the extent to which you can use the features of BI Server, such as the ability to publish and retrieve items and schedule the items you publish.

Using the user and group structure provided by your BI Server administrator, you can grant or deny access to the items you publish. If you find that you can't use features or perform activities appropriate to your work, ask your administrator to review the system permissions that have been assigned to you.

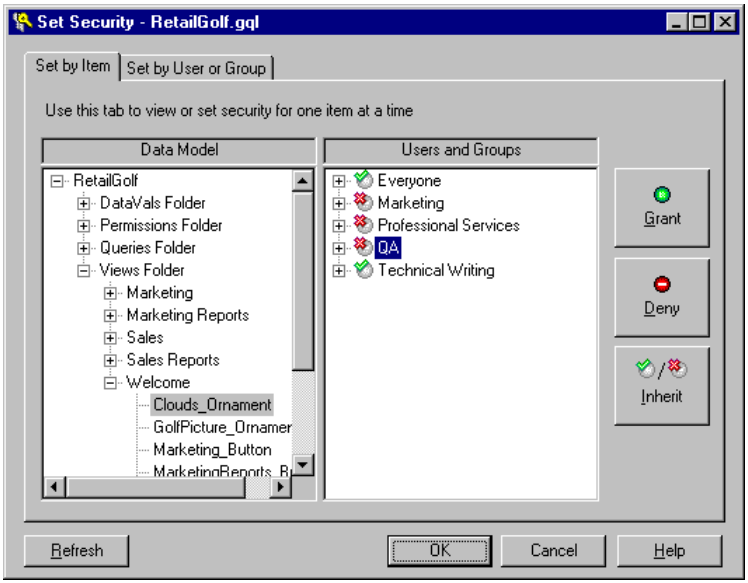
Access Inheritance Rules

In the BI Server security structure, users inherit group access according to the following rules:

Inheritance Rule	Description
Group members inherit their group's access.	When you grant or deny access to a group, all members of the group inherit that setting (if a group has access, each member has access and if a group is denied access, each member is denied it).
Members of more than one group inherit from the groups that grant access.	If a user belongs to one group that is granted access and to another group that is denied access, the user has access.
Member security settings override inherited group settings.	You can give a group member a setting different from that of the member's group. When you do, the overriding setting always applies to that member, even when you make it a member of other groups. You can remove an overriding setting to let a member inherit from the group again.

About the Set Security Dialog Box

You can set security for data models, queries, reports, and other data model components using the Set Security dialog box. This dialog box is available from any BI Query dialog box that lets you save or publish items. In BI Query Admin, you can also open the dialog box by clicking Set Security on the Tools menu.



The Set Security dialog box organizes the components of the data model into the following folders:

DataVals Lists the data values files associated with the data model.

Permissions Lists the data model permissions (such as Save Data Model and Edit Data Model).

Queries Lists the queries associated with the data model.

Views Lists the data model design windows and the data objects, object qualifiers, attributes, relationships, buttons, and ornaments that the windows contain.

You can set security for the individual items within the folders, although you can't directly grant or deny access to the folders themselves; they inherit the access granted or denied to the data model. You also can't deny yourself access. The symbols that indicate access settings in the Set Security dialog box are as follows:

Symbol	Description
	Green light. Access specifically granted.
	Red "no entry" symbol. Access specifically denied.
	Green check mark. A granted access that has been inherited.
	Red x. A denied access that has been inherited.

Setting Security

You can set security using one of the following methods:

By Item In this case, you select the item, then grant or deny access to that item for particular users and groups.

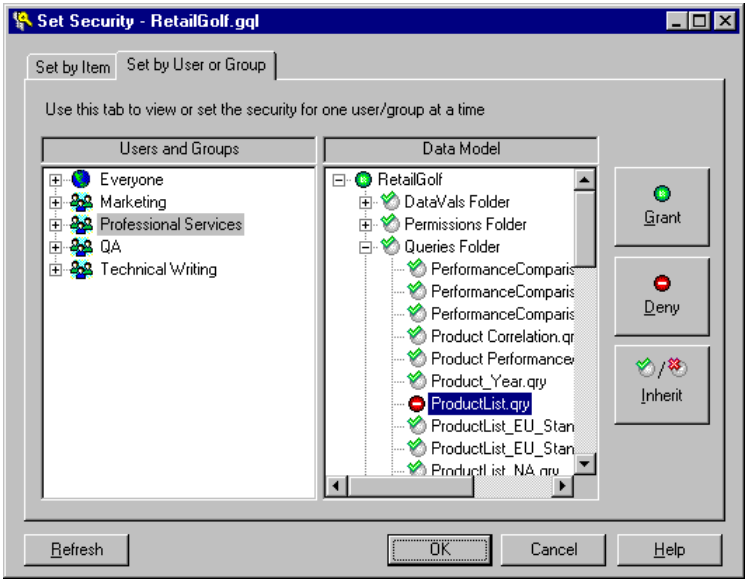
By User and Group In this case, you select a user or a group first, then select the items and grant or deny access to those items.

To set security:

1 Open the Set Security dialog box. The Set By Item page opens by default.

2 To set security by item, select the item from the Data Model list. From the User and Groups list, select the users and groups for whom you want to grant or deny access.

3 To set security by user and group rather than by item, click the Set By User and Group tab.



From the Users and Groups list, select the user or group for whom you want to grant or deny access. From the Data Model list, select the items that you want to secure.

4 Do one of the following:

- To grant access, click Grant.
- To deny access, click Deny.
- To allow a user or group to inherit security from the group of which it is a member, click Inherit.

5 When you have set security for as many items and as many users and groups as you want, click OK.

6 Publish the item.

Setting Security for Queries

When you publish data models, you can include queries with them and set security for the queries as part of setting security for the data model as a whole.

If you have the appropriate permissions, you can also add a query to a previously published data model, whether the model is your own or someone else's. When you publish such a query, it automatically has the security settings of the data model with which it's published.

If you're working in BI Query Admin, once you've published a query, whether as part of a data model or separately, you can change its security settings.

To set security for an individual query:

1 If the query has not been published, create or open it.

2 On the Query menu, click Publish. The Publish Query dialog box opens.

3 Do one of the following:

- To secure an unpublished query, type its name in the Query Name box.
- To secure a published query, select it from the Published Queries list.

4 Click Set Security.

5 In the Set Security dialog box, review the security assignments currently in effect for the query. Make any changes as necessary. For more information, see "Setting Security" on page 246.

6 When you've finished setting security, click OK.

7 Click Publish to publish the query.

Using BI Server Scheduler

If your querying environment includes the BI Server Repository and you have the appropriate publishing and scheduling permissions assigned by the BI Server administrator, you can schedule queries.

Scheduling regulates the flow of information from the database by processing the enterprise's queries and reports in a managed time frame. Jobs can be scheduled to run once or repeatedly, at specific intervals, during less busy times, or when a specific event has occurred. Scheduling also provides a variety of ways to distribute information.

For more information on default connection files, see the *BI Query Data Models User's Guide*.

If the queries in a data model you publish will be scheduled to run automatically, you must set a default connection file for the model before you publish it. Since the Scheduler runs queries from BI Server, not from the machine where they were created or last saved, you need to ensure as well that the connection you use is supported from the BI Server machine. Your BI Server administrator can provide you with this information.

For more information on scheduling, see the *BI Server Scheduling User's Guide*.

Scheduling a Query

You must publish your query before you can schedule it, and in order to publish a query, you must first publish the data model on which it is based.

To schedule a query:

1 On the Query menu, click Retrieve.

2 In the Retrieve Query dialog box, select a query to schedule and click Retrieve.

3 On the Query menu, click Schedule.

4 In the Schedule Job dialog box, provide the appropriate information. (The Schedule Job dialog box is part of another BI program module with its own Help system. For information about this dialog box, click its Help button.)

Retrieving Scheduled Results

For more information on scheduled queries, see the *BI Server Scheduling User's Guide*.

You can retrieve from the BI Server Repository any results sets generated by scheduled queries. To do so, you must have the BI Server Schedule Queries permission.

To retrieve results for a scheduled query:

1 On the Results menu, point to Retrieve, then click Scheduled Results.

2 In the Retrieve Results Sets dialog box, browse for and select a results set.

3 Click OK.

Appendix A

Accessibility and Technical Support

Accessibility **253**
Microsoft Accessibility Options 254

Technical Support **254**

Accessibility

Hummingbird products are accessible to all users. Wherever possible, we developed our software using Microsoft Windows interface standards. Depending on the product in which you are working, a subset of the following accessibility features is available.

Access Keys Menus have associated access keys (mnemonics). To open any menu, press Alt and the underlined letter in the menu name as it appears on the interface. For example, to access the File menu, press Alt+F.

Once you have opened a menu, you can access an item on the menu by pressing the underlined letter in the menu item name, or you can use the arrow keys to navigate the menu list.

Keyboard Shortcuts Some often-used menu options also have shortcut (accelerator) keys. The shortcut key for an item is listed beside it on the menu.

Directional arrows Use directional arrows on the keyboard to navigate through menu items or to scroll vertically and horizontally. You can also use the directional arrows to navigate through multiple options. For example, if you have a series of radio buttons, you can use the arrow keys to navigate the possible selections.

Tab key sequence To navigate through a dialog box, press the Tab key. Selected items appear with a dotted border. You can also press Shift+Tab to go back to a previous selection within the dialog box.

Spacebar Press the Spacebar to toggle check boxes on and off or to select buttons in a dialog box.

Esc Press the Esc key to close a dialog box without implementing any new settings.

Enter Press the Enter key to select the highlighted item or to close a dialog box with the new settings. You can also press the Enter key to close all About boxes.

ToolTips ToolTips appear for functional icons. This feature lets users use screen reviewers to make interface information available through synthesized speech or through a refreshable braille display.

Microsoft Accessibility Options

Microsoft Windows environments contain accessibility options that let you change how you interact with the software. This feature can add sound, increase the magnification, and create sticky keys.

To access the Microsoft Windows Accessibility options, open Control Panel and click Accessibility.

If you installed the Microsoft Accessibility components for your Windows system, you can also find other Accessibility tools on the Start menu under Programs/Accessories/Accessibility.

To add the Accessibility components:

1 In Control Panel, double-click Add/Remove Programs.

2 On the Setup tab, select the Accessibility Options check box and click Apply.

3 Click OK.

Technical Support

Administrator's can contact Hummingbird Technical Support to report problems or suggest enhancements. We require product and company information before we can investigate any problems, so please fill-out the electronic form on the Hummingbird web site or phone in/fax us the corresponding information before contacting Technical Support.

For Technical Support services, please use the contact information for your area.

Hummingbird Ltd. 1 Sparks Avenue, North York, Ontario, Canada M2H 2W1	
North America	Tel: 1 800 486 0095 (8:00 am - 8:00 pm EST) Fax 1 613 548 7616 Email: hbi-support@kingston.hummingbird.com
Europe (Except UK and Scandanavia)	Tel: +33 1 55 35 96 80 (9:00 am-6:00 pm GMT + 1hr) Fax: +33 1 42 61 31 87 Email: fr-bi-support@hummingbird.com
UK and Scandanavia	Tel: +44 (0)118-902-9507 (9:00 am-5:00 pm GMT) Fax: +44 (0) 118 978 9325 Email: uk-support@hummingbird.com
Other Locations	+1 613 548 4355 (8:00 am-8:00 pm EST) +1 613 548 7616 Email: hbi-support@kingston.hummingbird.com

Using the Trace Utility

Hummingbird provides a trace utility with the software to help troubleshoot problems you are having. The trace utility simplifies problem-solving by monitoring the activity of your products. If you are having problems with the software, Technical Support may ask you to run the trace utility, reproduce the problem, save the trace information, and send us the resulting trace file.

To run the trace utility, double-click `trace.exe` in the following path:

```
Program Files\Hummingbird\BI\Utility
```

For information of configuring the trace utility, see Trace Help.

Index

Symbols

!= operator 47
formatting character 65
% wildcard character
    in qualifications 46
&& text string 171
&«prompt» text string 170
&d text string 170
&dn text string 170
&p text string 170
&pp text string 170
&q text string 170
&t text string 170
* formatting character 65
*= operator 206
<= operator 45, 47
<> operator 45
= operator 45, 47
=* operator 206
> operator 45, 47
>= operator 45, 47
\ formatting character 65
_ formatting character 65
_ wildcard character 46

A

aborting queries 30
accessibility features 253
ad hoc
    drill-down mode 127
    queries 7, 27
Ad Hoc Drill-Down Mode 127
administrator
    BI Query 3
    role 3
administrator-layer files 20
    location 20
    split data model 8
aggregate functions 179
    and grouping 179
    applying to attributes 182
    AVERAGE 180, 181
    AVERAGE DISTINCT 181
    COUNT 181
    COUNT DISTINCT 181
    defined 179
    example of grouping with 179
    grouping after calculating 181
    grouping before calculating 182
    MAXIMUM 181
    MINIMUM 181
    SUM 181
    SUM DISTINCT 181
aliased data 87
    data values file 88
aliases for data objects (correlation names) 210
alphabetic sort order 136
"and/or" qualifications 41
ANSI SQL/89 outer-join syntax 206
ANSI SQL-92 outer-join syntax 206
appending rows 141
applications
    exporting results to, overview 227
    using buttons to export results to 227

applying
aggregate functions to attributes 182
operators to qualifications.....38
Attribute Name box 37
attribute window
checking available data values 39
creating queries.....27
grouping attributes 182
modifying SQL 209
qualification tree.....36
sort box 32
sorting rows 32
attributes
calculated, creating 188
calculated, overview.....186
definition 5
in variables 114
placeholders in variables.....115
qualifying 36
qualifying other attributes 44
selecting 28
automatic variables 113
Automatically Order Outer Joins 202
permission 203
preference 203
AVERAGE aggregate function.....180
AVERAGE DISTINCT aggregate function .181

B

.bak files.....53
batch conversion, standard reports to
BI Query Reports 174
Begins with (operator) 46
match any/match all 48
BETWEEN (operator) 45, 48
BI Query
overview.....3
administrator.....3
disconnect from DBMS.....23
opening a data model 18
Reports.....8
starting 18
BI Query Admin 3
administrator..... 3
BI Query Reports..... 8
BI Query Update 3
BI Query User 3
BI Server..... 12
environment..... 18
log on..... 21
Repository 12
system permissions 13
terminology..... 13
BI Server permissions
Publish Queries to Admin Layer 239
Publish Queries to User Layer 239
Schedule Queries..... 250
BI Server security..... 243
access inheritance rules 244
access symbols..... 246
securing queries..... 248
setting by item 246
setting by user or group 246
users and groups 244
BIGroupID 113
using in a query qualifier..... 121
BIUserID..... 113
using in a query qualifier..... 121
borders
adding to standard reports..... 168
building
circular queries without correlations... 215
queries 28
buttons
combining results sets in Excel 233
defined..... 7
export options 224
exporting results to applications..... 227

C

calculated attributes
overview 186
choosing functions..... 189
creating..... 188
qualifying other attributes..... 44

calculated columns
in standard reports ... 160
calculating a function
grouping after ... 181
grouping before ... 182
calculations ... 179
calculations, in standard reports
adding grand totals ... 159
basing one on another ... 160
in report columns ... 160
cancelling
qualifications ... 36
queries ... 30
changing
column order in results ... 137
super queries ... 132
characters
comparing ... 47
inserting/deleting in results/reports ... 65
matching patterns ... 46
spaces appearing as nulls ... 49
strings containing wildcards ... 46
checking
conditional prompts ... 105
data values ... 39
group prompts ... 103
query specifications ... 33
super queries ... 135
chronological sort order ... 136
circular queries ... 210
building with a correlation ... 213
outer joins in ... 216
with correlations ... 210
without correlations ... 214, 215
Classic (ANSI SQL/89) outer-join syntax ... 206
clearing previous queries ... 30
column headings
in standard reports ... 152
columns
adjusting size in standard reports ... 162
banding in standard reports ... 164
changing heading style
in standard reports ... 166
combining in standard reports ... 163
hiding in results ... 137
joining ... 143, 146
reordering ... 31
reordering in results ... 137
reordering in standard reports ... 137
specifying order ... 31
stacked by the Report Converter ... 173
stacking in standard reports ... 165
combined data models ... 8, 10
combined queries
exporting ... 33
saving ... 33
combining
columns ... 146
qualifications ... 42
results sets ... 140, 141
three results sets in Excel ... 233
commands
Dynamic Data Exchange (DDE) . 232, 233
common language names
substitute for database names ... 87, 88
comparing
characters ... 47
dates ... 47
values ... 47
component queries
deleting ... 134
editing in super queries ... 133
Condition box ... 37
conditional prompts
overview ... 93
creating ... 105
naming ... 105
testing ... 105
connecting
to a database ... 22
to a DBMS ... 22

connection files
changing outer-join syntax....206
connecting to a DBMS 22
default 22
using non-default 22
connections
database 22
DBMS 22
Contains (operator)
match any/match all 48
converting
single standard reports 174
copying results for export 219
correcting
errors in super query windows....135
queries....53
correlation names for data objects (alias)....210
correlations
circular queries without....214
COUNT aggregate function 181
Count All
modifier 43
COUNT DISTINCT aggregate function....181
creating
calculated attributes....188
conditional prompts....105
data values files to alias data 88
data values query files 85
data values results files.... 82, 83
dynamic relationships....198
group prompts....103
prompts....101
queries....28
subqueries....191
variables 118
variables, overview....114
creating tables
in the database 196
curly braces....103
curly brackets, for text entry 75
customizing
results for users....121
customizing data models 8

D

data
types for prompts 94
data alias
overview 87
creating a data values file 88
data display
results, form view 125
spreadsheet view.... 125
data formats 57
converting year formats 74
correcting invalid data entry formats.... 75
data entry formats 72
data type formats.... 57
date entry format.... 74
formatting options 58
in standard reports.... 153
modifying SQL.... 207, 208
number-entry format.... 73
opening the Format dialog box.... 61
order of formatting options 59
predefined formats.... 58
target types 60
data formats, options
category-specific options 58
Computed String option.... 59
Null values option 59
order of options 59
String option 58
data formats, specifying
default formats 63
for boolean data 68
for data entry.... 73
for numeric data.... 69
for years (number of digits) 74
the Computed String option.... 66
the Null Values option 64
the String option 65
data model design 3

data models
customizing ... 8
definition ... 5
local ... 18
opening ... 18
passwords ... 18
Recent ... 18
repository ... 19
retrieving ... 19
split ... 20
types ... 8
data models, combined ... 8, 10
data models, split ... 8
data objects
definition ... 5
in circular queries ... 210
querying ... 50
relating ... 197, 198
using twice in one query ... 210
data sets
overview ... 192
character delimiters in ... 193
comments in ... 193
file extensions ... 192
in qualification trees ... 194
.qrr files ... 192
syntax ... 193
using as a qualifier ... 194
using .qrd files as a data set ... 195
viewing ... 195
data type formatting categories ... 57
data values
checking ... 39
load all ... 39
qualifying with ... 39
data values files ... 79
overview ... 79
aliasing data ... 88
data aliasing ... 87
in conditional prompts ... 105
naming ... 80
naming for use in prompts ... 80
order of use ... 81
storing ... 80
types ... 80
use in prompts ... 81
using in queries ... 86
data values query files
creating ... 85
defined ... 80
location ... 33
naming ... 85
saving ... 33
data values results files
creating ... 82
creating with a text editor ... 83
defined ... 80
naming ... 82
using as a data set ... 195
data, determining if missing ... 49
database
overview ... 4
attributes ... 4
connection files ... 22
creating tables from query results ... 196
fine-tuning the SQL ... 209
functions in variables ... 117
modifying SQL ... 208
records ... 4
supported operators ... 50
date formats
ambiguous ... 74
automatic conversion of ... 74
dates
comparing ... 47
DBMS connections ... 4, 22
DBUserID ... 113
using in a query qualifier ... 121
DDE
combining Excel commands ... 231
combining three results sets
in Excel ... 233

DDE commands
combining....232
Excel....233
new....233
paste....232
run....233
default
data values query....39, 79
export options, overriding....224
export options, setting....224
results options....221
default text ornaments....169
deleting
component queries....134
dynamic relationships....201
operations in super queries....134
results....227
values in qualification tree....36
Design Mode....11
design windows
definition....6
disconnecting from DBMS....23
displaying
data values....39
prompt values in queries....107
Distinct modifier....43
Does not begin with (operator),
match any/match all....48
Does not contain (operator),
match any/match all....48
Does not contain (operator),
matching character patterns....46
Does not end with (operator),
match any/match all....48
duplicates
eliminating....43, 185
dynamic relationships....197
between unrelated data objects....197, 198
choosing operators....197
creating....198
deleting....201
outer joins in....198

E

editing
component queries of super queries....133
operations in super queries....134
super queries....132
variables....118
eliminating duplicates....43, 185
e-mail, exporting BI Query files....229
e-mail, exporting information....229
Ends with (operator)
match any/match all....48
matching character patterns....46
entering
data values....39
equijoins....144
error messages
modifying SQL....207, 208
errors in super query windows....135
Excel
combining commands with DDE....231
combining three results sets....233
setting export options for....230
using two commands....231
excess joins, cleaning up....51
excluding
rows with character spaces....49
rows with null values....49
executive buttons....7
export options....223, 224
Excel....230
exporting
BI Query files by e-mail....229
queries and results sets....219
SQL to text files....230
exporting, results
copy and paste....219
options....220
preparations....219
to an application....227
using e-mail....229

F

file extensions for data sets....192

files
administrator-layer 20
.bak 53
connection 22
data values 79, 80
data values query 33, 80
data values results 80
.gqa 8
.gql 10
.gqu 8
.qrd 80, 226
.qrr 80, 226
.qry 33, 80
query 33
.rep 174
results 226
.rpt 174
filtering
results 136
results sets, super queries 129
finding
items in query results 125
ranges of values 48
fine-tuning SQL 209
footers in standard reports 152
formatting characters
..... 65
functions
AVERAGE 180
AVERAGE DISTINCT 181
COUNT 181
COUNT DISTINCT 181
MAXIMUM 181
MINIMUM 181
SUM 181
SUM DISTINCT 181
functions, in calculated attributes 189

G

.gqa files 8
.gql files 10
.gqu files 8
grand totals, adding to standard reports 159
group prompts
creating 103
naming 103
order of execution 103
overview 92
testing 103
grouping
after calculating functions 181
before calculating functions 182
with aggregate functions 179
without calculating functions 185

H

headers in standard reports 152
hiding columns in results 137
Hummingbird accessibility 253

I

importing
prompts 106
variables from a file 120
IN operator 45, 47
including
database functions in variables 117
prompts in variables 117
rows with character spaces 49
rows with null values 49
inserting
database functions in variables 117
prompts in queries 102
subqueries 191
variables into variables 117
invalid data formats, correcting 75
IS NOT NULL qualification operator 49

J

joining
columns 143, 146
rows 141
tables 197, 198

joins
columns 143, 146
conflicts 204
natural joins 145
operators for join conditions 197
order in outer joins 201
removing unnecessary from queries 51
types 144

L

layers
administrator 8
user 8
left outer joins 144
legacy join operators 206
lists, matching items 47
loading
queries 35
results 126
local data model 18
logging on to BI Server 18, 21

M

macro sheets, opening automatically 231
MAPI, mailing from BI Query 229
matching
all values 48
any values 48
character patterns 46
items in lists 47
materialized relation, definition 204
MAXIMUM aggregate function 181
merged tables in outer joins 202
MINIMUM aggregate function 181
missing data 49
modifiers
Count All 43
Distinct 43
query 43
Trim Relationships 43
modifying
component queries 132, 133
.qrd file with a text editor 83
queries 35, 43
SQL 207, 208
super queries 132
multi-pass reporting 160

N

naming
attributes and operators in variable 114
conditional prompts 105
data values files 80
data values files used in prompts 80
data values query files 85
data values results files 82
group prompts 103
prompts 94, 101
natural joins 145
negative
qualifications 42
nested queries 190
creating 191
inserting 191
new outer-join syntax 206
non-SQL operators 50
NOT BETWEEN (operator) 45, 48
NOT IN (operator) 45, 47
not indicator 42
NULL
operator 49
values, format order 59
null values 49
numeric sort order 136

O

ODBC
syntax for outer joins 206
offline, working 18
opening
data models 18
super query windows 132

operations
deleting in super queries 134
editing in super queries..... 134
operator list
qualification tree 45
operators
!= 47
*=..... 206
< 45, 47
<=.....45, 47
<>..... 45
= 45, 47
=*..... 206
> 45, 47
>=.....45, 47
applying to qualifications..... 38
Begins with 46, 48
BETWEEN 45, 48
Contains 48
database supported 50
Does not begin with 48
Does not contain.....46, 48
Does not end with 48
Ends with.....46, 48
for join conditions..... 197
IN.....45, 47
in variables..... 114
IS NOT NULL..... 49
NOT BETWEEN.....45, 48
NOT IN 45, 47
NULL..... 49
placeholders in variables 115
User defined 50
options
export, setting for Excel 230
results, actions of..... 220
ordering columns in results..... 137
ornaments
in standard reports..... 169
special text strings in 170
outer join types
full outer joins.....144
left outer joins.....144
right outer joins 144
outer joins
available syntax 206
Classic (ANSI SQL/89) syntax 206
conflicts.....204
in circular queries 216
in dynamic relationships 198
in queries 201, 206
join order 201
new syntax 206
syntax, changing per connection.....206
using legacy joins as is 53
outer joins, join order 201
Automatically Order Outer Joins
preference and permission.....203
Show Join Order preference.....201
specifying join order.....203

P

passwords
BI Server log on 21
data model 18
paste, DDE command.....232
performing calculations 114
using variables 114
permissions
Automatically Order Outer Joins 202, 203
placeholders
examples of use in variables 116
for attributes and operators.....115
preferences
Automatically Order Outer Joins 202, 203
Preserve Legacy Query Outer Joins. 52, 53
results options.....220
Show Join Conflicts 204
Show Join Order..... 201, 202
prefixes, SQL 209
Preserve Legacy Query Outer Joins preference . 52, 53
printing standard reports.....157

prompts
overview 91
and curly braces 103
appearance in BI Web 91
automatic formatting 95
conditional, creating 105
conditional, overview 93
conditional, testing 105
creating 101
data types 91, 94, 95
data values allowed, BI Web 97
data values in BI Web 97
default values 91
display objects in BI Web 98
examples of uses 99
group, creating 103
group, overview 92
group, testing 103
importing 106
in query data objects 91
in variables 117
inserting in queries 102
naming and storing 101
naming conventions 94
no formatting 96
optional versus required 96
showing values in queries 107
specifying web-reporting values 107
types 92
using curly braces 102
using more than once in queries 102
using radio buttons 98
publishing
data values results files 243
queries 239
standard reports specifications 241
publishing queries 239
deleting published queries 241
requirements for 239

Q

.qrd files 80, 226
.qrr files 80, 226
.qry files 33, 80
Qualification Edit box 37
qualification operators
!= 47
< 47
<= 47
<> 45
= 47
> 47
>= 47
Begins with 46
BETWEEN 45
Does not contain 46
Ends with 46
IN 45
IS NOT NULL 49
NOT BETWEEN 45
NOT IN 45
NULL 49
User defined 50
qualification tree
combining qualifications 42
modifying values 36
negative qualification 42
operator list 45
using 36
qualifications
combining 42
multiple 42
multiple with "and/or" 41
negative 42
restricting with operators 45

qualifying
attributes 36, 38
attributes with prompts 102
attributes with variables 119
more than one attribute with a prompt 102
one attribute with another 44
queries 38
queries, overview 36
with data values 39
with data values files 86
with lists 47
with ranges of values 48
with subqueries 191
with user name variables 121
queries 27
overview 27
ad hoc 7, 27
building 28
building with a correlation 213
cancelling 30
circular 210
circular with correlations 210
circular without correlations 214, 215
clearing previous 30
combining results 140
correcting 53
default data value 79
definition 7
exporting 33, 219
exporting the SQL 230
loading and modifying 35
modifying for selected results 127
multiple, combining 140
nested 190
outer joins in 201, 206
qualifying 36, 38
qualifying with variables 119
refreshing 52
removing unnecessary joins 51
retrieving and loading 35
reusing 35
running 29
saving 27, 33
saving results 226
submitting 29
too long 30
using legacy joins as is 53
viewing specifications 33
query
data objects containing prompts 91
file location 33
multiple data objects 50
results, creating tables from 196
viewing contents 33
viewing specifications 33
Query Admin
administrator 3
query modifiers
Count All 43
Distinct 43
Trim Relationships 43
query results, searching 125
querying multiple data objects 50

R

range of results
narrowing 139
range of values, finding 48
real database name
using to qualify attributes 44
recent data model 18
record, defined 4
reducing network load 196
refreshing queries 52
Regional Options (Control Panel) 58
Regional Settings (Control Panel) 58
relational database management system (RDBMS) 4
relationships 197
between unrelated data objects 198
missing from data objects 197
new 197
trim 51

reordering
columns 31
columns in results....137
standard reports columns....137
.rep files 174
Report Converter utility....173
installing....173
troubleshooting 173
reports
BI Query 8
BI Query Reports....8
Standard Reports....8
Repository 12
repository data model 19
restricting
qualifications with operators....45
result sets, exporting 219
results
overview....125
BI Server environment 125
combining results sets 140, 141
definition 7
deleting 227
files 226
filtering, overview 136
finding text....125
form view....125
grouping after calculating functions 181
modifying queries for
selected results....127
narrowing the range 139
opening....126
preparing for export 219
saving 226
set 125
sorting....136
sorting, before or after submitting 136
spreadsheet view....125
using buttons to export
to applications....227
results options
actions of 220
button preferences 221
preferences 220
Save Column Headings.... 220
Save Prompts.... 220
Save Query Name.... 220
specifying for export 221, 224
Text in Quotes.... 220
Transpose 221
results section in standard reports 152
results sets
combining 141
combining, overview.... 140
definition 125
super queries 129
results, exporting
by copying and pasting 219
options.... 220
to an application 227
retrieving
data model.... 19
queries 35, 241
values using variables 114
reusing queries.... 35
reverse qualifications 42
right outer joins 144
rows
appending.... 141
database 4
sorting 31
.rpt files 174
Run Mode.... 11
run, DDE command.... 233
running
queries 29

S

Save Column Headings results option.... 220
Save Prompts results option.... 220
Save Query Name, results option.... 220

saving
combined queries 33
data values queries 33
dynamic relationships 198
queries 33
results 226
subqueries 33, 191
scalar functions 189
scheduling queries by using variables 114
searching, query results 125
selecting
operators for joins 197
separators in results options 220
server 18
setting
export options 224
export options for Excel 230
Show Join Conflicts preference 204
Show Join Order preference 201, 202
showing
prompt values in queries 107
query specifications 33
sort order
attributes 32
rows 31
specifying 136
sorting
alphabetical or chronological 136
before submitting 31, 32
results 136
results, before or after submitting 136
reverse order 136
rows 31
special text strings
for standard reports 170
specifications
applying in standard reports 156
for standard reports 155
publishing 241
specifying
column order 31
export options 224
export options for Excel 230
sort order 136
sort order of results 136
split data models
overview 8
administrator layer 8, 20
user layer 8
spreadsheet view of results 125
SQL
exporting to text files 230
fine-tuning 209
modifying 208
modifying, overview 207
permissions required to modify 207
prefixes and suffixes 209
stacked columns, in BI Query Reports 173
Standard Reports 151
applying specifications 156
converting multiple reports 174
converting to BI Query Reports 174
creating 153
data formats in 153
default text ornaments in 169
exporting reports to text files 157
installing the Report Converter 173
printing 157
publishing specifications 241
reordering columns 137
report specifications 154
saving specifications 155
special text strings for 170
structure and characteristics 152
subheadings in 164

Standard Reports, editing
adding borders....168
adding calculated columns....160
adding grand totals....159
adding subtotals....158
adding text ornaments....171
adjusting column size....162
banding columns....164
basing one calculation on another....160
changing heading text style....166
combining columns....163
customizing column settings....167
editing default ornaments in....172
formatting columns....166
stacking columns....165
starting queries....30
submitting queries....29
subqueries
creating....191
data types....190
exporting....33
inserting....191
invalid operators....192
overview....190
saving....33
substituting
common language names....87, 88
subtotals, adding to standard reports....158
suffixes, SQL....209
SUM aggregate function....181
SUM DISTINCT aggregate function....181
super queries....129
overview....129
cannot export SQL....230
combining multiple results sets....129
correcting errors....135
editing....132
editing component queries....133
filtering results sets....129
from filtering results....136
modifying component queries....132
super query windows....129
testing....135
super query windows
opening....132
overview....129

T

tables
creating in the database....196
database....4
joining....197, 198
Technical Support....254
Teradata DBMS, prompt data types....95
terminology....5
testing
conditional prompts....105
group prompts....103
super queries....135
text
matching in queries....46
ornaments, special strings in....170
text files
exporting SQL to....230
Text in Quotes results option....220
text ornaments
in standard reports....169
text strings
&&....171
&«prompt»....170
&d....170
&dn....170
&p....170
&pp....170
&q....170
&t....170
three-tier environment....18
Transpose results option....221
trim relationships....51
Trim Relationships modifier....43
types of column joins....144

U

updating
database....3
queries....52

User defined operators 50
user layer, split data model 8
user name, BI Server log on 21

V

values
    comparing 47
    null 49
variables
    overview 113
    attribute and operator information 114
    BIGroupID 113, 121
    BIUserID 113, 121
    creating 114, 118
    DBUserID 113, 121
    editing 118
    examples of placeholder use 116
    importing from a file 120
    in queries converted to QDO 113
    in scheduled queries 114
    including database functions 117
    including prompts 117
    naming 113
    naming attributes and operators 114
    performing calculations 114
    placeholders for attributes and operators 115
    retrieving values 114
    storage 113
    uses 114
    using to qualify attributes 119
    within variables 117
viewing
    data sets 195
    query specifications 33

W

web-reporting values 107
wildcard characters 46
windows
    Attribute 5
    super query 132
wiping queries 30
working
    offline 18, 21
    online 21

Y

years, two digit to four digit 74